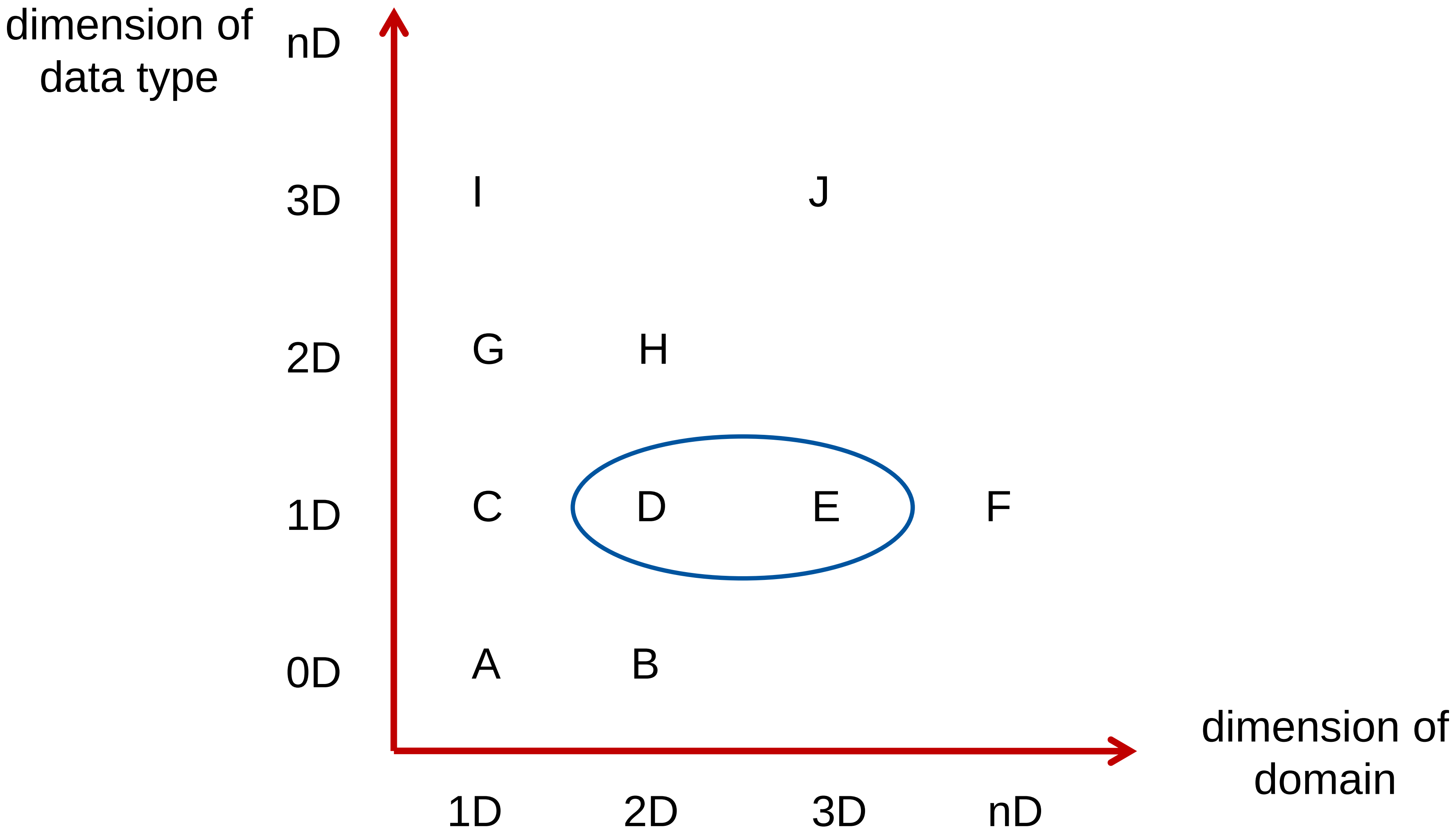
Scalar Field Visualization

Leif Kobbelt

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Types of Data



Function Plots and Height Fields

- Visualization of 1D or 2D scalar fields
  - 1D scalar field: $\Omega \subset \mathbb{R} \to \mathbb{R}$
  - 2D scalar field: $\Omega \subset \mathbb{R}^2 \to \mathbb{R}$

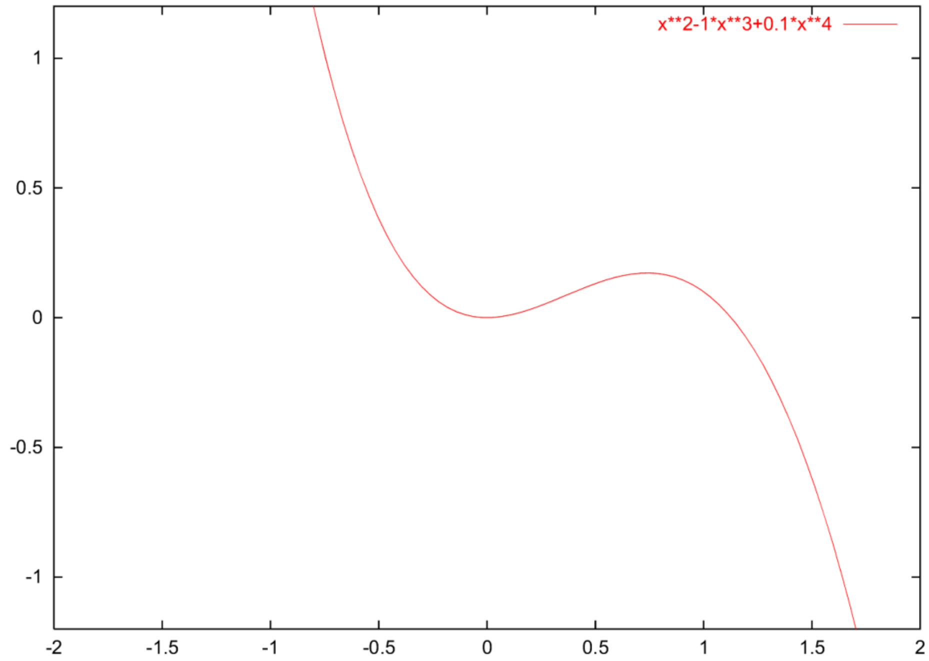
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Function Plots and Height Fields

- Function plot for a 1D scalar field
  - 2D Points

    $$\{(s, f(s)) \mid s \in \mathbb{R}\}$$

  - 1D manifold: line

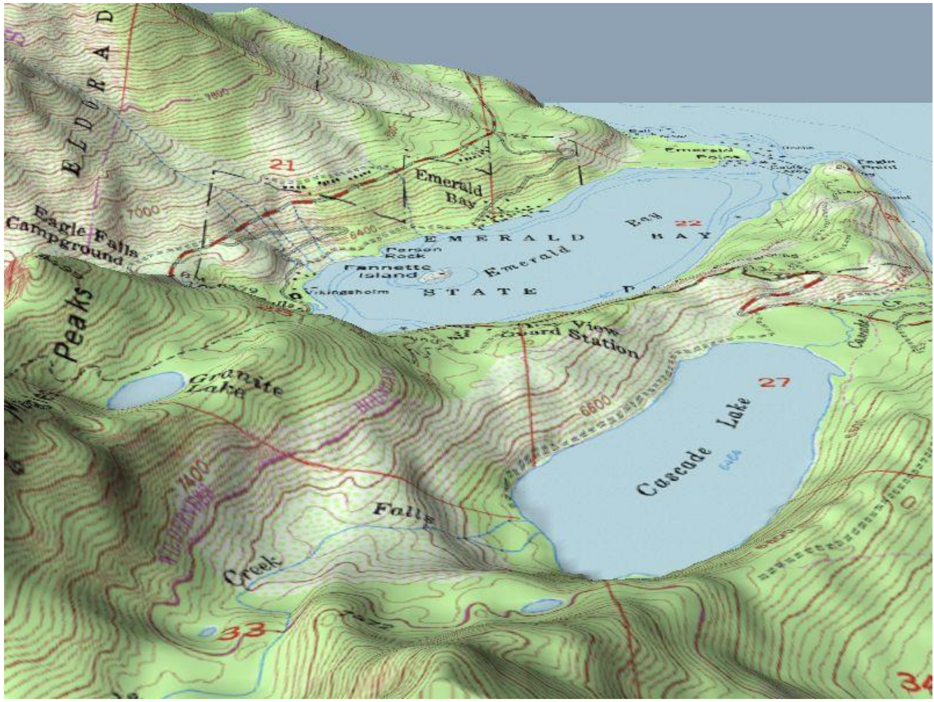


Function Plots and Height Fields

- Function plot for a 2D scalar field
  - 3D Points

$$\{(s,t,f(s,t) \mid (s,t) \in \mathbb{R}^2\}$$

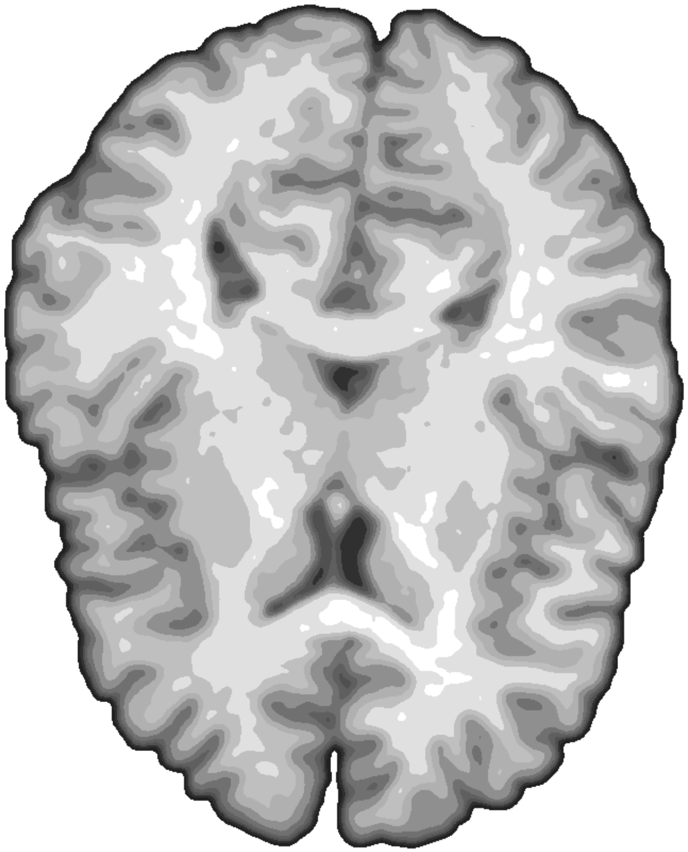
  - 2D manifold: surface



- Surface representations
  - Wireframe
  - Hidden lines
  - Shaded surface

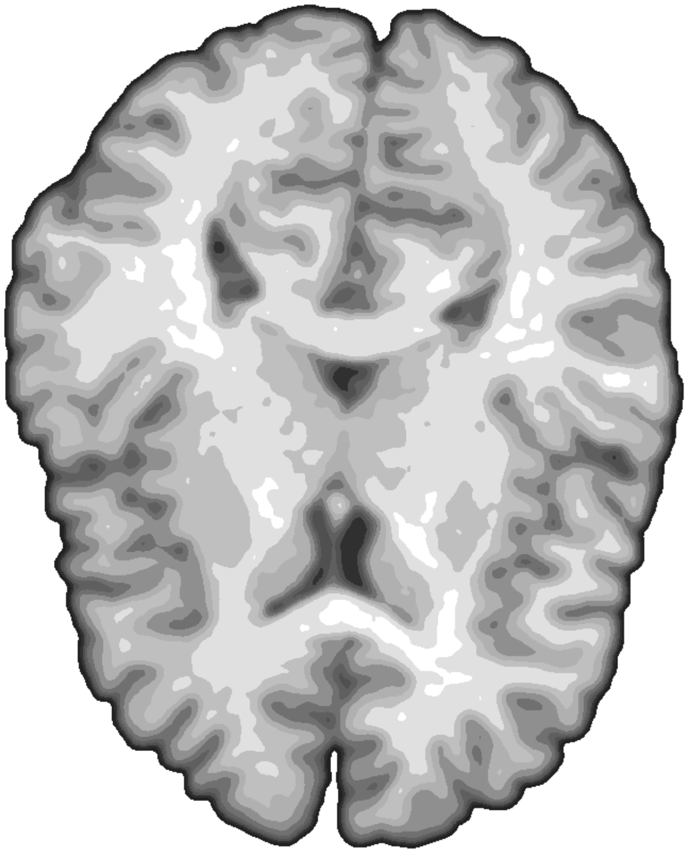
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

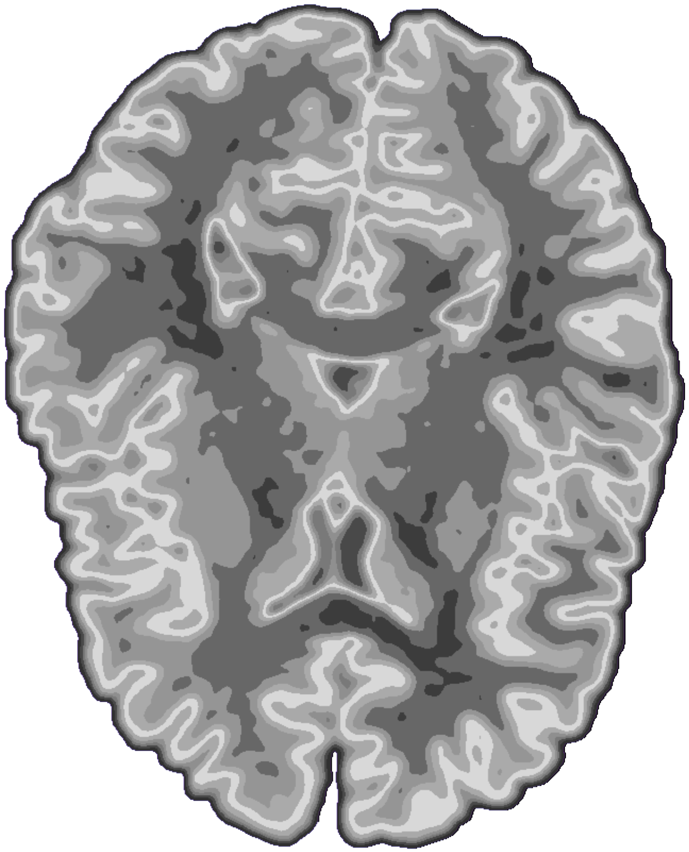
„The Geometry of Images“



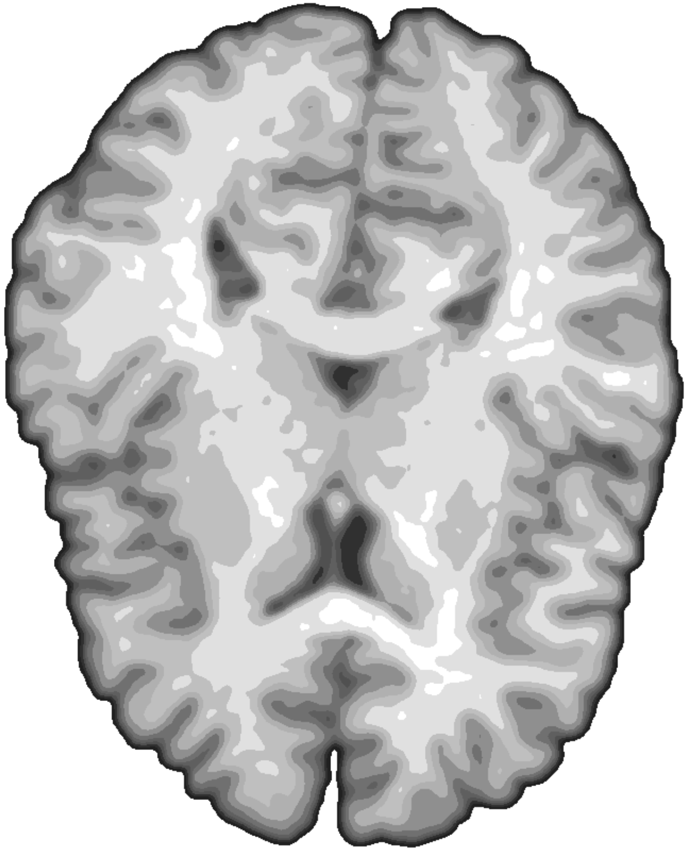
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

„The Geometry of Images“





„The Geometry of Images“



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Isolines

- Visualization of 2D scalar fields
- Given a scalar function $f : \Omega \mapsto I\!R$

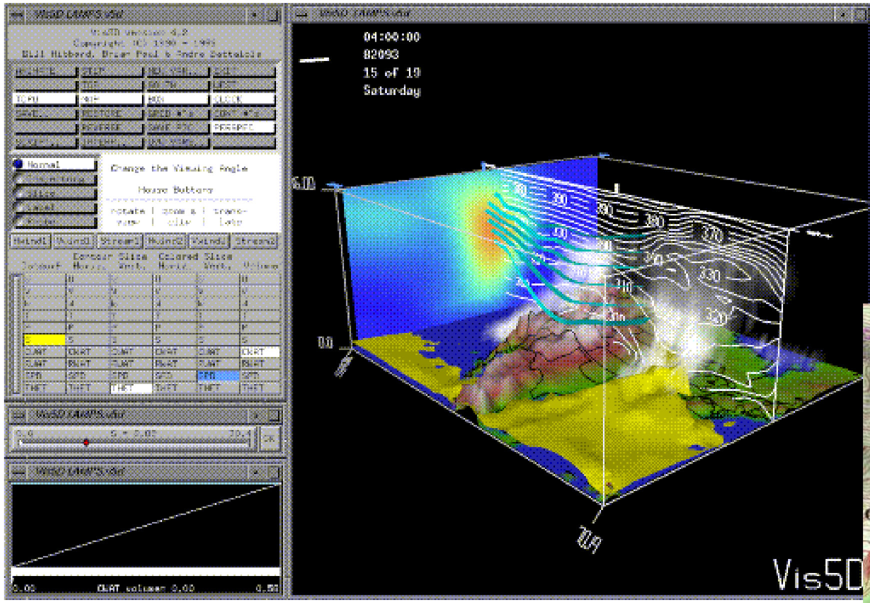
  and a scalar value $c \in R$
- Isoline consists of points

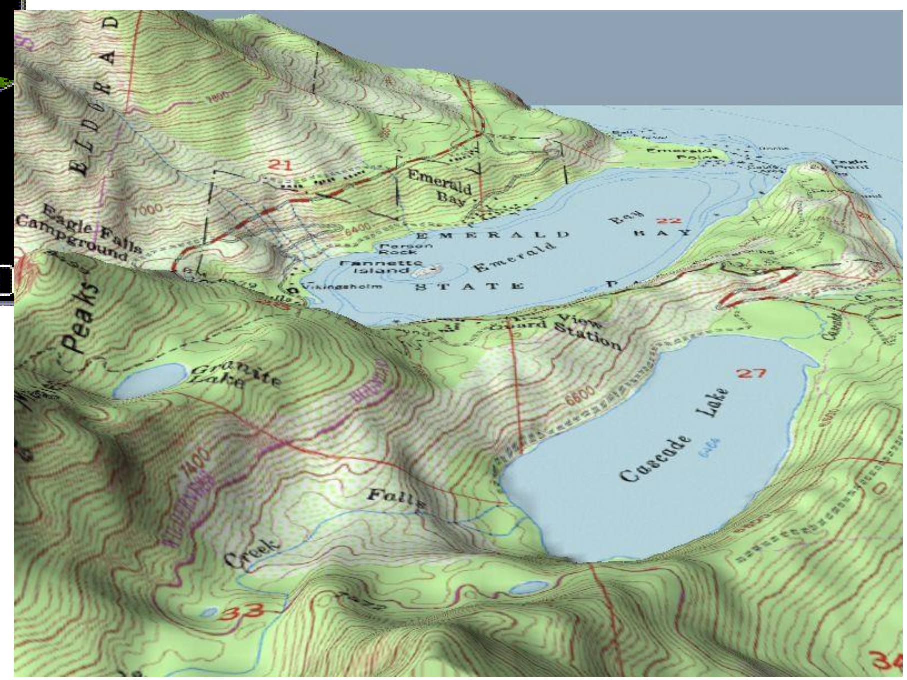
$$\{(x, y) \mid f(x, y) = c\}$$

- If *f()* is differentiable and *grad(f)* $\neq$ 0, then isolines are curves

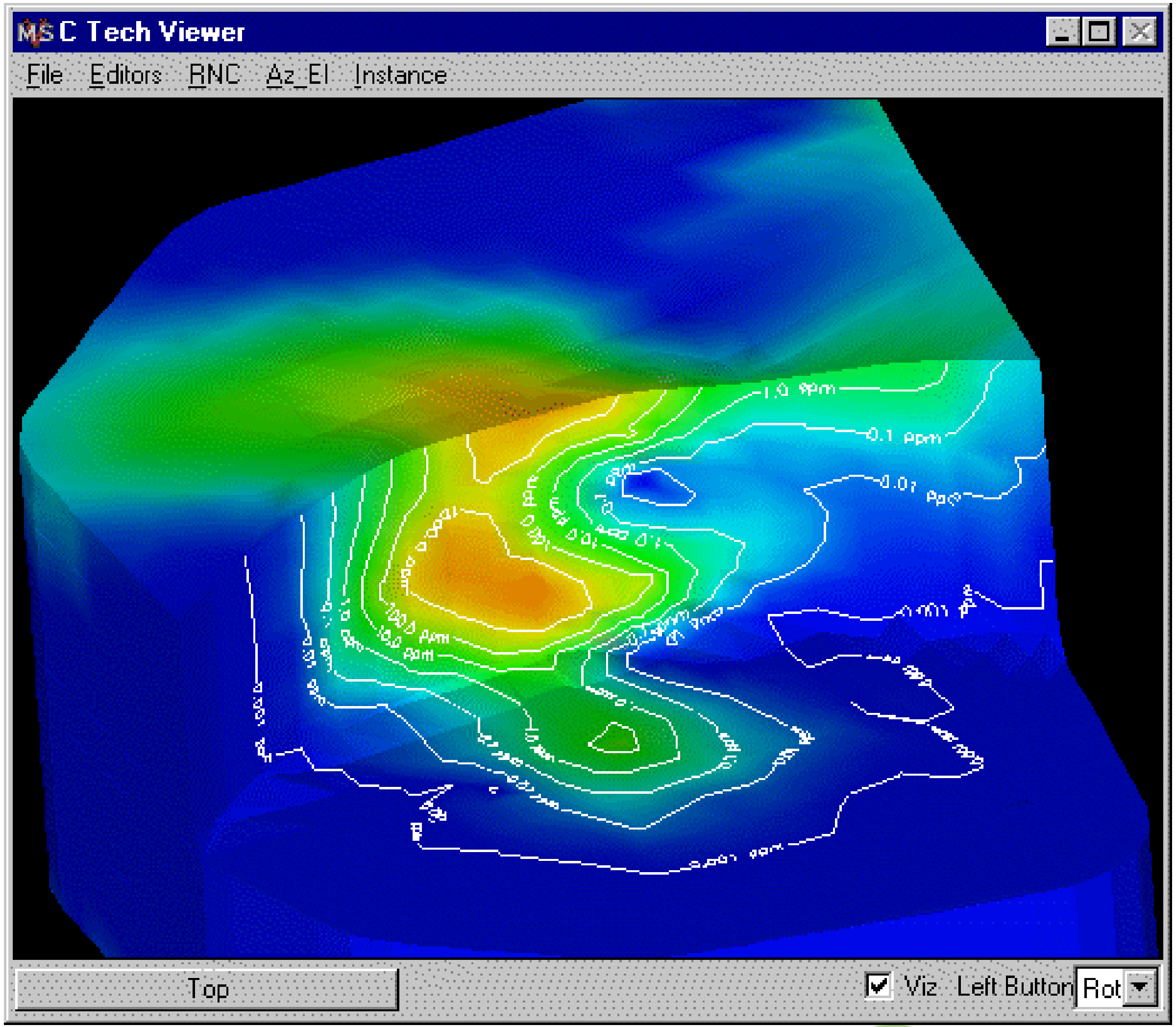
- Contour lines

Isolines


04:00:00
82093
15 of 19
Saturday
Vis5D


Emerald Bay
Eagle Falls Campground
Granite Lake
Cascade Lake
Peaks
Falls
Creek
21
22
27
33

Isolines


C Tech Viewer
File Editors RNC Az El Instance
1.0 ppm
0.1 ppm
0.01 ppm
Top
Viz Left Button Rot

Isolines

- pixel by pixel contouring
- straightforward approach: scanning all pixels for isovalue
- input
  - $f : (1,\ldots,x_{max}) \times (1,\ldots,y_{max}) \rightarrow \mathbf{R}$
  - Isovalues $I_1,\ldots, I_n$ and isocolors $c_1,\ldots,c_n$
- algorithm

  `for all` $(x,y) \in (1,\ldots,x_{max}) \times (1,\ldots,y_{max})$ `do`

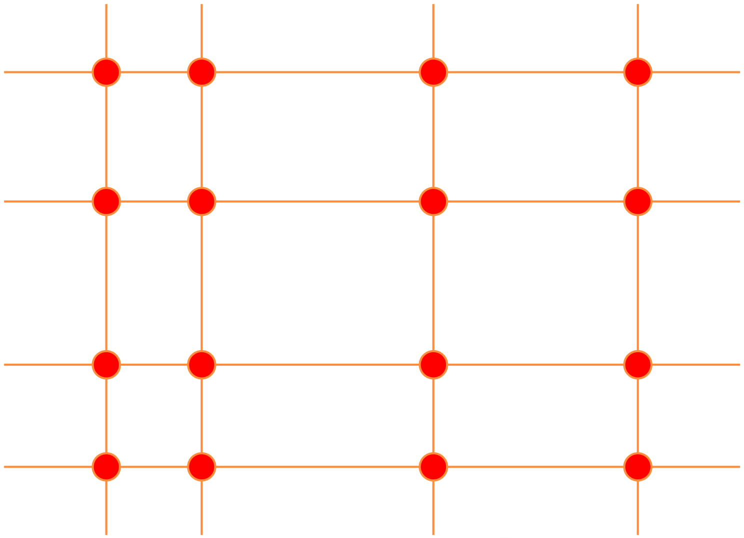
    `for all` $k \in \{1,\ldots,n\}$ `do`

      `if` $|f(x,y)-I_k| < \varepsilon$ `then`

        `draw(`x,y,c_k`)`
- problem: isolinie can be missed if the gradient of *f()* is too large (despite range ε)

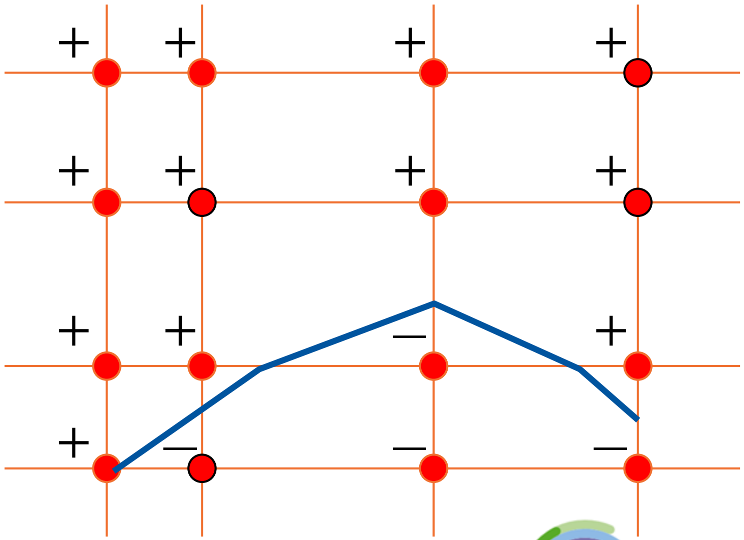
Isolines

- Marching squares
  - representation of the scalar function on a rectilinear grid
  - scalar values are given at each vertex $f \leftrightarrow f_{ij}$
  - take into account the interpolation within cells
  - isolines cannot be missed
  - divide and conquer: consider cells independently of each other



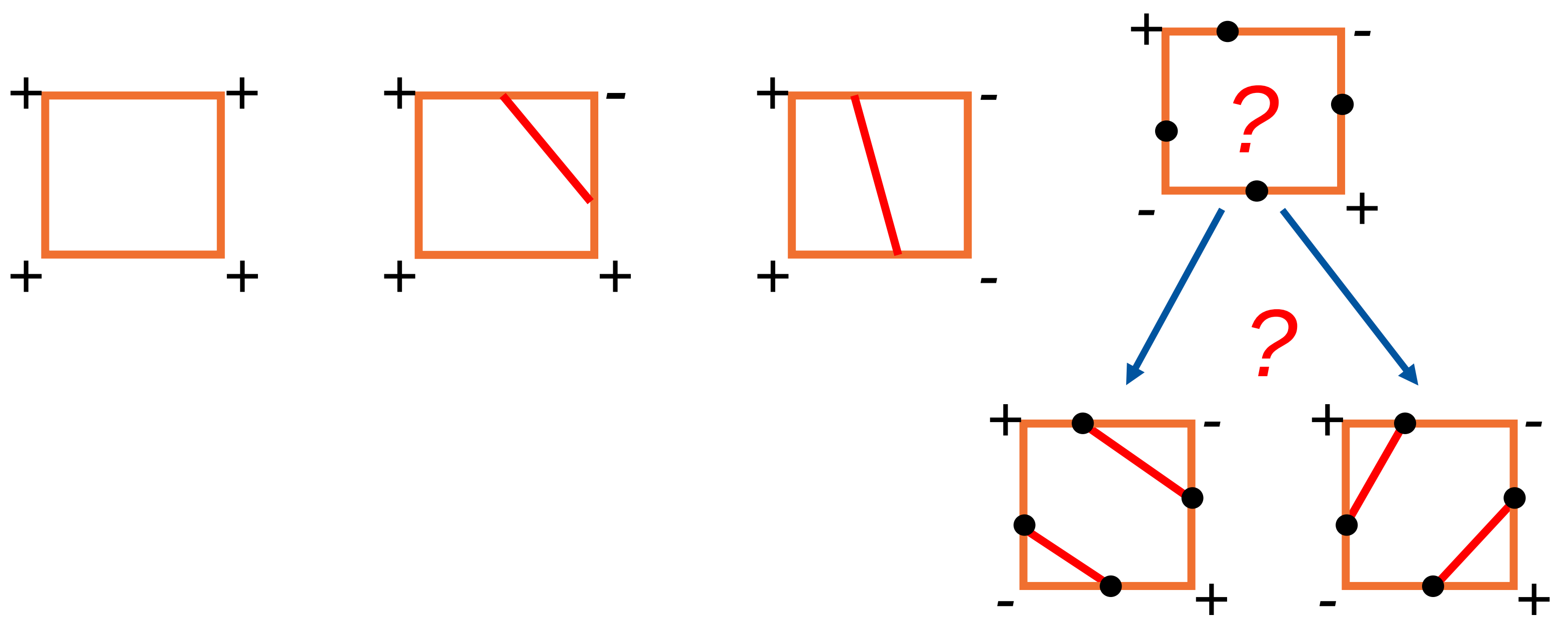
Isolines

- Which cells will be intersected ?
  - Initially mark all vertices by + or – , depending on the conditions $f_{ij} \geq c$, $f_{ij} < c$
- No isoline passes through cells (=rectangles) which have the same sign at all four vertices
  - So we only have to determine the edges with different signs



Isolines

- Only 4 different cases (classes) of combinations of signs
- Symmetries: rotation, reflection, change + ↔ -
- Compute intersections between isoline and cell edge, based on linear interpolation along the cell edges



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

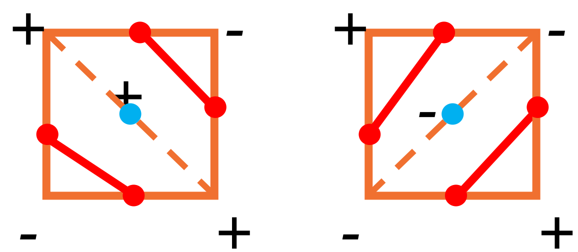
Isolines

- We can distinguish the cases by a decider

- Mid point decider
  - Interpolate the function value in the center

$$f_{\text{center}} = \frac{1}{4}\left(f_{i,j} + f_{i+1,j} + f_{i,j+1} + f_{i+1,j+1}\right)$$

  - If $f_{center} < c$ we chose the right case, otherwise we chose the left case

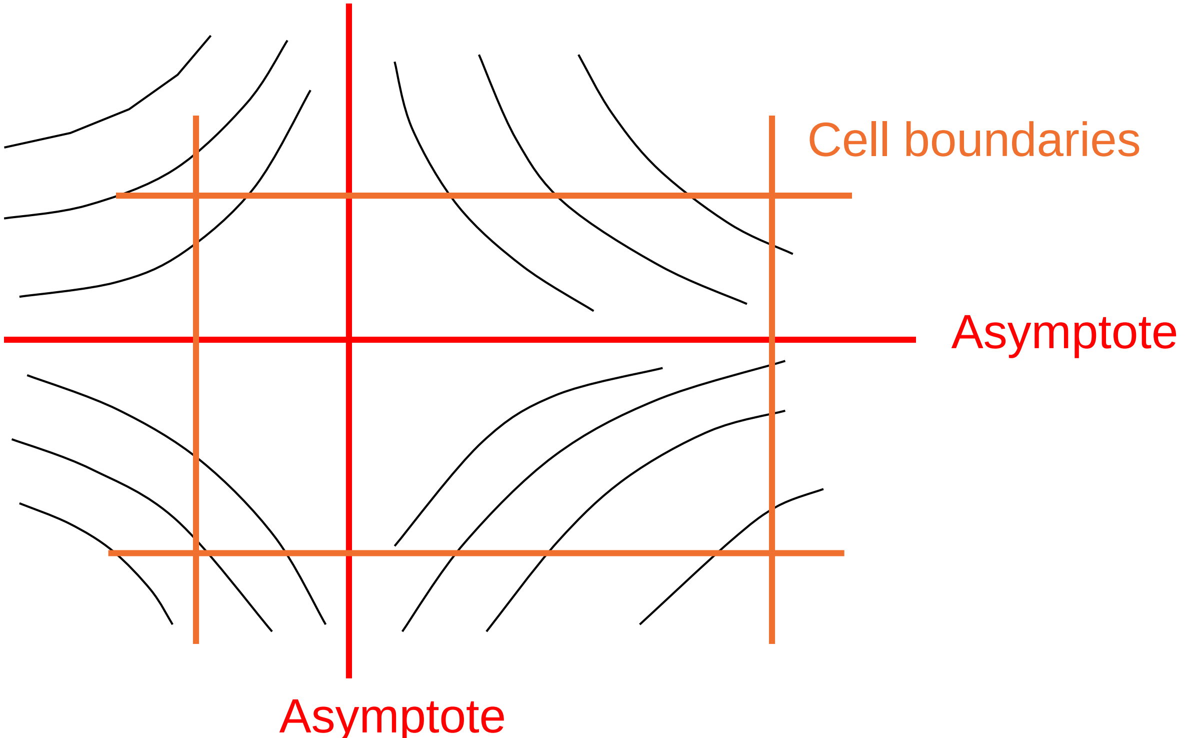


  - Not always correct solution

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Isolines

- Asymptotic decider
  - Consider the bilinear interpolant within a cell
  - The true isolines within a cell are hyperbolas



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Isolines

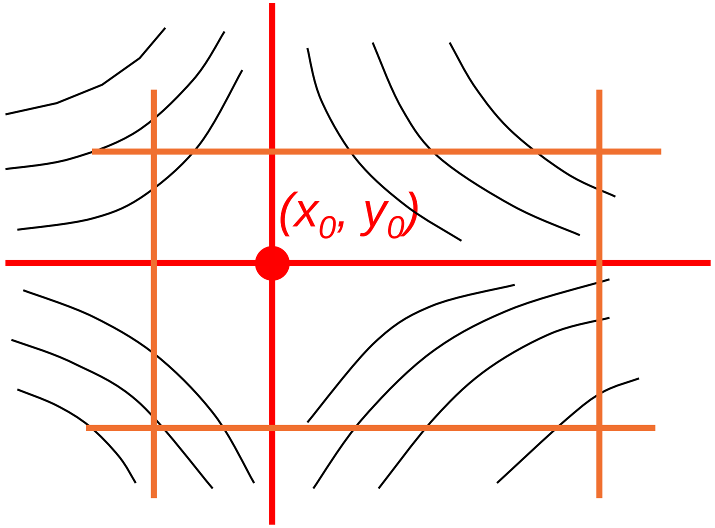
- Interpolate the function bilinearly

$$f(x,y) = f_{i,j}(1-x)(1-y) + f_{i+1,j}x(1-y) + f_{i,j+1}(1-x)y + f_{i+1j+1}xy$$

- Transform *f()* to

$$f(x,y) = \alpha(x-x_0)(y-y_0) + \beta$$

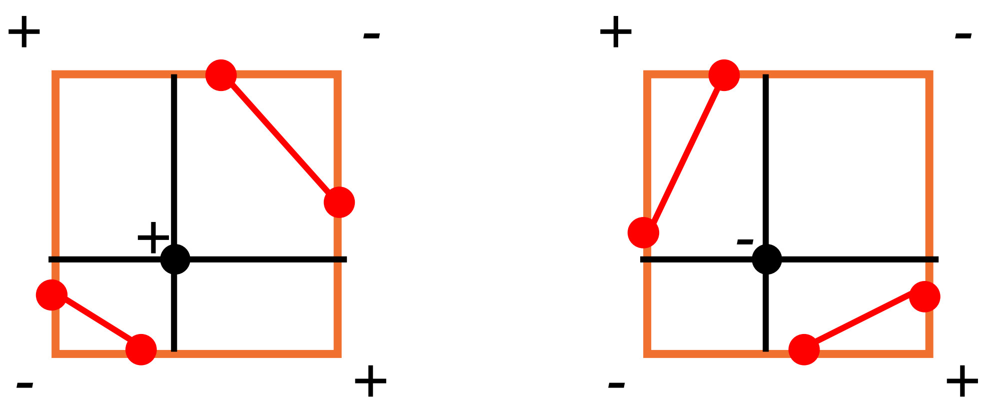
- β is the function value in the intersection point of the asymptotes



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization
Visual Computing Institute
RWTH AACHEN UNIVERSITY

Isolines

- If $\beta \leq c$ we chose the right case, otherwise we chose the left one

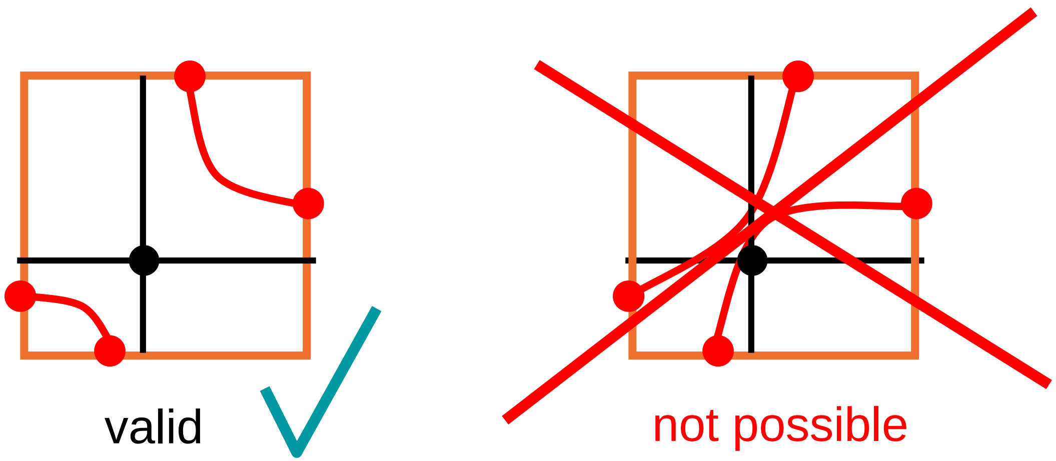


Isolines

- Explicit transformation *f()* to

$$f(x,y) = \alpha(x - x_0)(y - y_0) + \beta$$

  can be avoided
- Idea: investigate the order of intersection points either along x or y axis
- Build pairs of first two and last two intersections



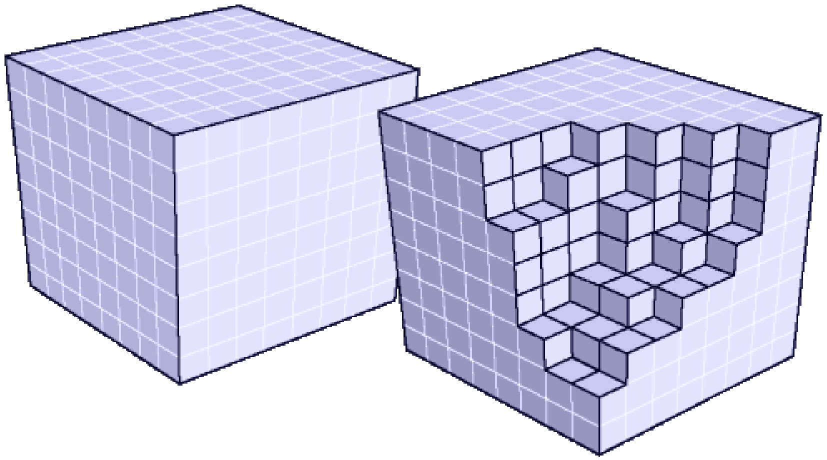
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

3D Scalar Field Visualization

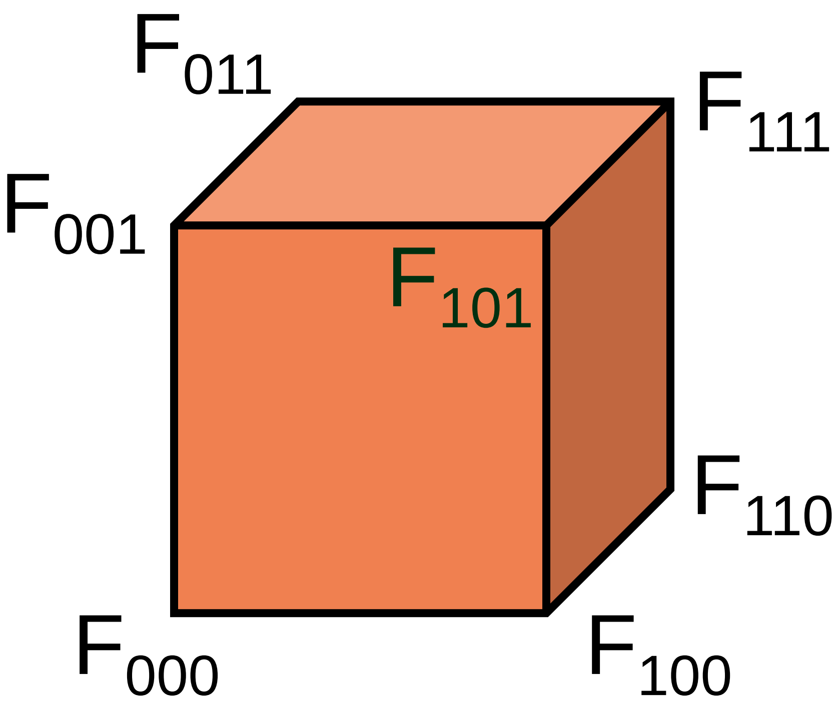
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volumetric Representations

- Implicit representation $F(x,y,z) = 0$
- Signed distance function
- Sample on a uniform cartesian grid
- Trilinear interpolation



Volumetric Representations



$$
\begin{aligned}
F(i+u,j+v,k+w) = \\
F_{000}\ (1-u)(1-v)(1-w)\ + \\
F_{100}\ \quad u\ (1-v)(1-w)\ + \\
F_{010}\ (1-u)\quad v\ (1-w)\ + \\
\ldots \\
F_{111}\quad u \quad v \quad w
\end{aligned}
$$

Volumetric Representations

- *Indirect Rendering*:
  - Isosurface extraction: *Marching Cubes*

- *Direct Rendering*:
  - Ray Casting

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Visual Computing Institute
RWTH AACHEN UNIVERSITY

Volume Data

- discrete representation:

$$F_{ijk} = F(i\,\Delta x,\ j\,\Delta y,\ k\,\Delta z)$$

- piecewise polynomial (linear) interpolation

- tri-linear functions per cell („voxel“)
  $\Rightarrow$ algebraic iso-surfaces of degree 3

- approximation by a polygonal mesh

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Extraction

- Find point samples on the iso-surface
  - exploit voxel structure
  - use function values

- Connect neighboring samples
  - exploit voxel neighborhood structure
  - relation between voxels and polygons

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Sampling

in which voxel ?

Surface Sampling

in which voxel ?

$$F((i+u)\,\Delta x,\ (j+v)\,\Delta y,\ (k+w)\,\Delta z) =$$

$$\begin{aligned}
&(1-u)(1-v)(1-w)F_{ijk} &&+ (1-u)(1-v)\,w\,F_{ijk+1} &&+\\
&(1-u)\quad v\,(1-w)F_{ij+1k} &&+ (1-u)\quad v\ w\,F_{ij+1k+1} &&+\\
&\quad u\,(1-v)(1-w)F_{i+1jk} &&+ \quad u\,(1-v)\,w\,F_{i+1jk+1} &&+\\
&\quad u\quad v\,(1-w)F_{i+1j+1k} &&+ \quad u\quad v\ w\,F_{i+1j+1k+1} &&+
\end{aligned}$$

$$(u,v,w) \in [0,1]^3 \ \Rightarrow\ \min F_{ijk} \leq F(u,v,w) \leq \max F_{ijk}$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Sampling

in which voxel ?

$S_c[F]$ passes through all (and only) voxels with $\min F_{ijk} \leq c \leq \max F_{ijk}$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Sampling

in which voxel ?

$S_c[F]$ passes through all (and only) voxels with $\min F_{ijk} \le c \le \max F_{ijk}$

how to find these voxels efficiently ?

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Sampling

in which voxel ?

$S_c[F]$ passes through all (and only) voxels with $\min F_{ijk} \leq c \leq \max F_{ijk}$

how to find these voxels efficiently ?

- store min/max $\{F_{ijk}, F_{ijk+1}, F_{ij+1k}, F_{ij+1k+1}, F_{i+1jk}, F_{i+1jk+1}, F_{i+1j+1k}, F_{i+1j+1k+1}\}$
- build octree

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Min/Max Tree

- store min/max for every voxel V_{ijk}^{n}
- octree V_{ijk}^{n-1} has children

$$V_{2i\ 2j\ 2k}^{n} \cdots V_{2i+1\ 2j+1\ 2k+1}^{n}$$

- propagate min/max values upwards
- surface extraction:
  traverse only the relevant voxels

$$O(n^3) \Rightarrow O(n^2 \log n)$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

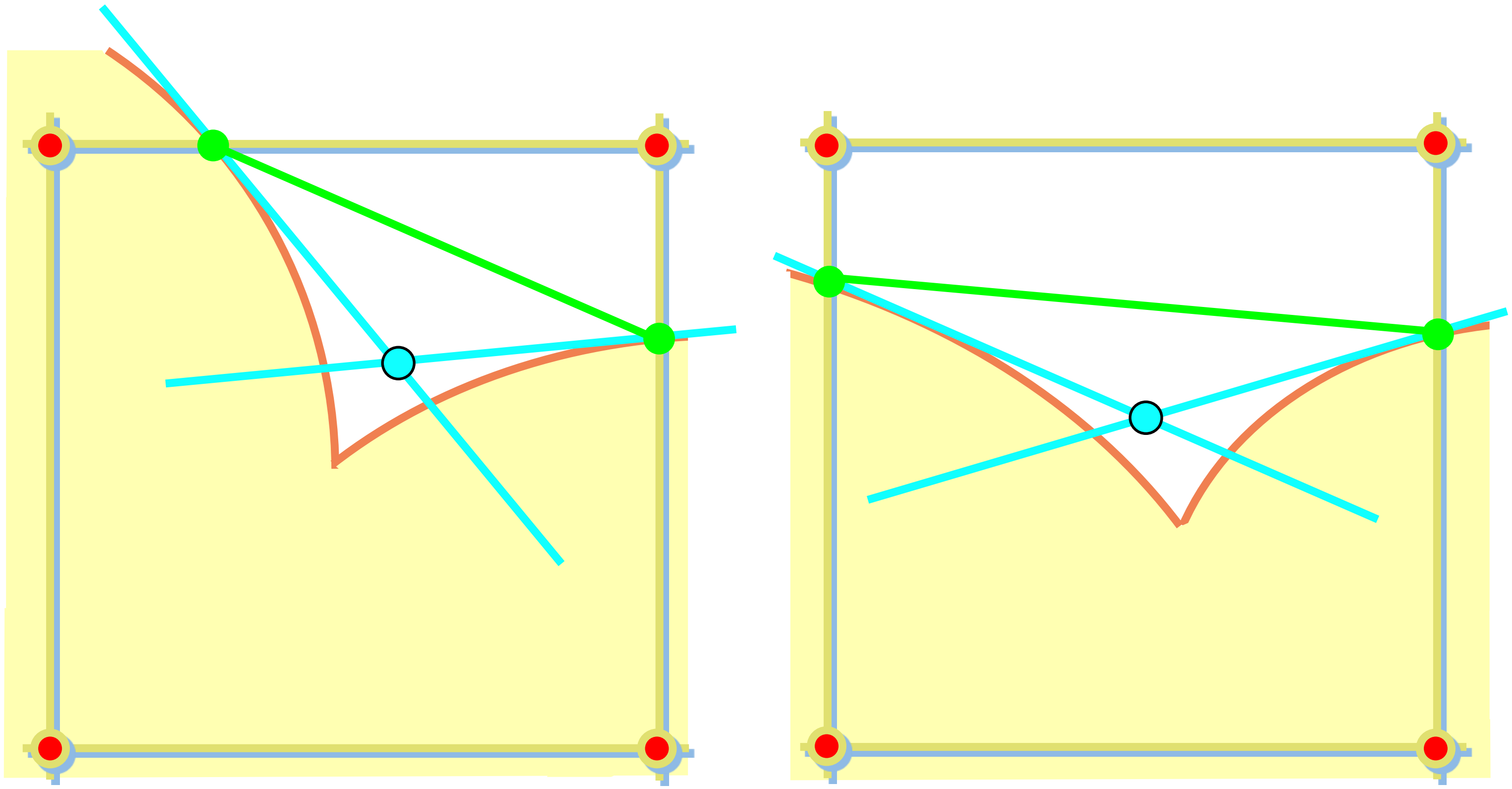
Surface Sampling

- per voxel : ray intersection
- ray ... $P + \lambda Q$
- intersection ... $G(\lambda) = F(P + \lambda Q) = c$
- solve cubic equation ... 

Surface Sampling

- per voxel : ray intersection
- ray ... $P + \lambda\, Q$
- intersection ... $G(\lambda) = F(P + \lambda\, Q) = 0$
- solve cubic equation ... 
- ray parallel to coordinate axis
  $\Rightarrow$ solve linear equation ... 

Problems / Bad Approximation



Surface Sampling

- use additional information to improve iso-surface approximation

- normal vector to iso-surface $S_c[F]$:

$$\nabla F(x,y,z) = [\ \partial F/\partial x,\ \partial F/\partial y,\ \partial F/\partial z\ ]$$

- find surface samples by (approximately) intersecting tangent planes

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Error Quadrics

- squared distance to plane:

$$v = (x, y, z, 1)^T, \qquad p = (a, b, c, d)^T$$

$$\text{dist}(p, v)^2 = (p^T, v)^2 = v^T (pp^T) v =: v^T Q_P v$$

$$Q_P = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

- sum distances to vertex' planes:

$$\sum_{\text{planes } p} \text{dist}(p, v)^2 = \sum_{\text{planes } p} v^T Q_p v = v^T \sum_{\text{planes } p} Q_p v =: v^T Q_v v$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Error Quadrics

- optimal point that minimizes the error:

$$\left(\sum_p n_p n_p^T\right) v^* = -\left(\sum_p d_p n_p\right) \qquad p = \left(\underbrace{a_p, b_p, c_p}_{=:n_p \text{ with } \|n_p\|=1}, d_p\right)^T$$

- solve using pseudo-inverse (SVD) (underdetermined vs. overdetermined)

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Surface Data

- polygon meshes / triangle meshes

  $$M = (\{p_i\},\{T_j\})$$

- compute samples
  - on grid edges
  - within voxels

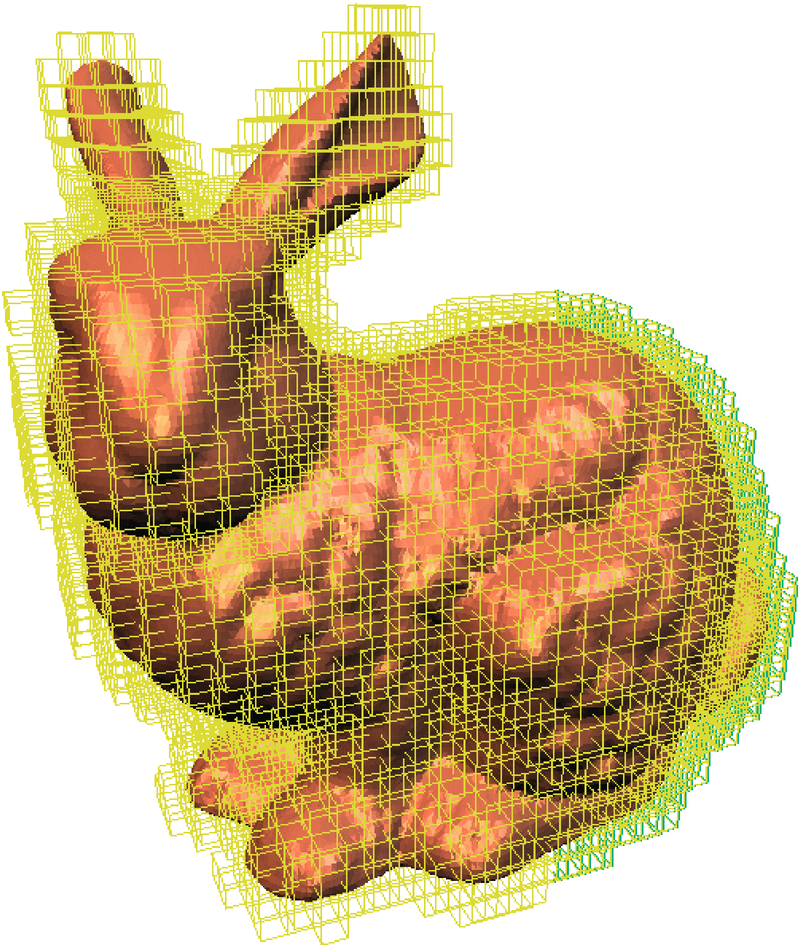
- compute faces (vertex connectivity)
  - for each voxel
  - for each grid edge

Marching Cubes [Lorensen/Cline ´87]

- extract surface patch for every grid cell
  - classification by the signs at the corners (black, white, gray)
- approximate samples
  - linear interpolation of *scalar* distance values along cell edges
- look-up table with pre-computed triangulations
  - 2^8 entries, many symmetries

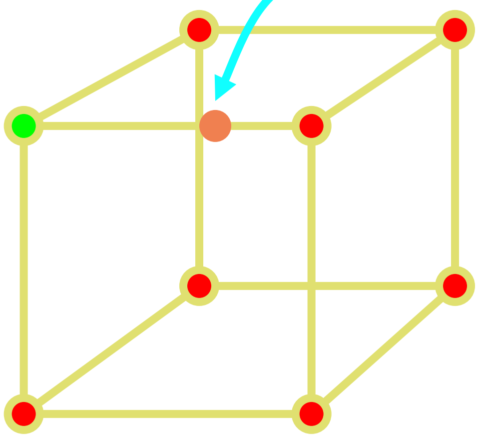
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Visit Each Gray Cell ...



Compute Samples on Edges

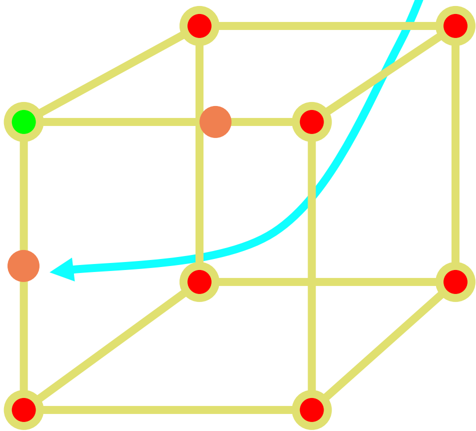
$$\underbrace{\left(i + \frac{\left| d_{i,j,k} \right|}{\left| d_{i,j,k} \right| + \left| d_{i+1,j,k} \right|}, j, k \right)}$$



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Compute Samples on Edges

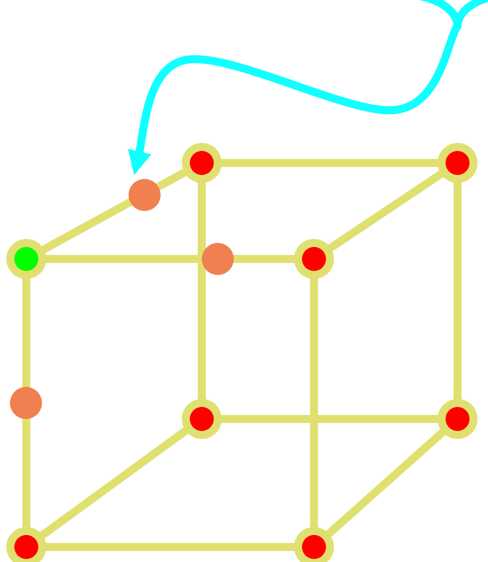
$$\underbrace{\left(i, j + \frac{\left|d_{i,j,k}\right|}{\left|d_{i,j,k}\right| + \left|d_{i,j+1,k}\right|}, k \right)}$$



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

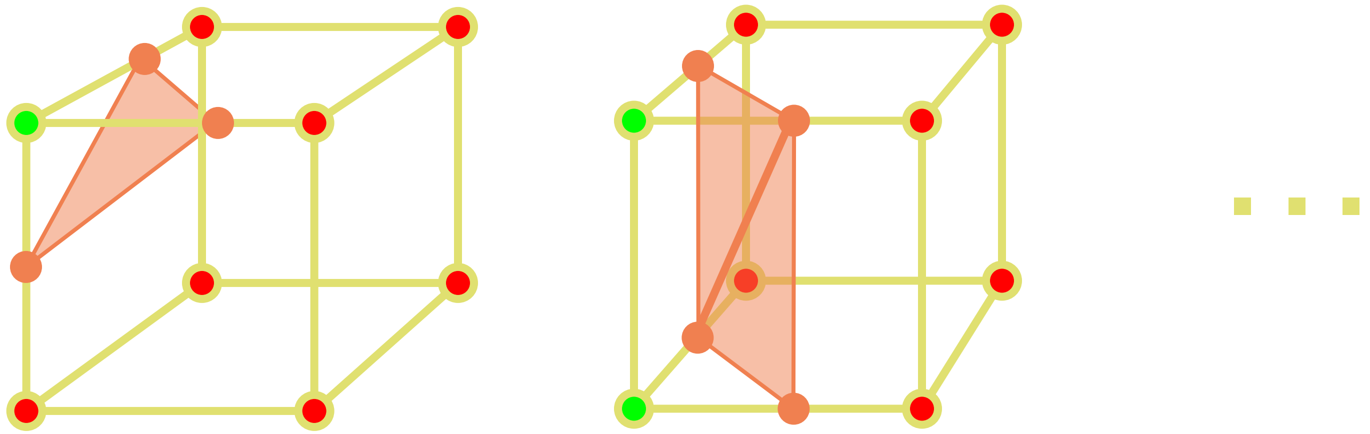
Compute Samples on Edges

$$\left(i, j, k + \frac{\left| d_{i,j,k} \right|}{\left| d_{i,j,k} \right| + \left| d_{i,j,k+1} \right|} \right)$$

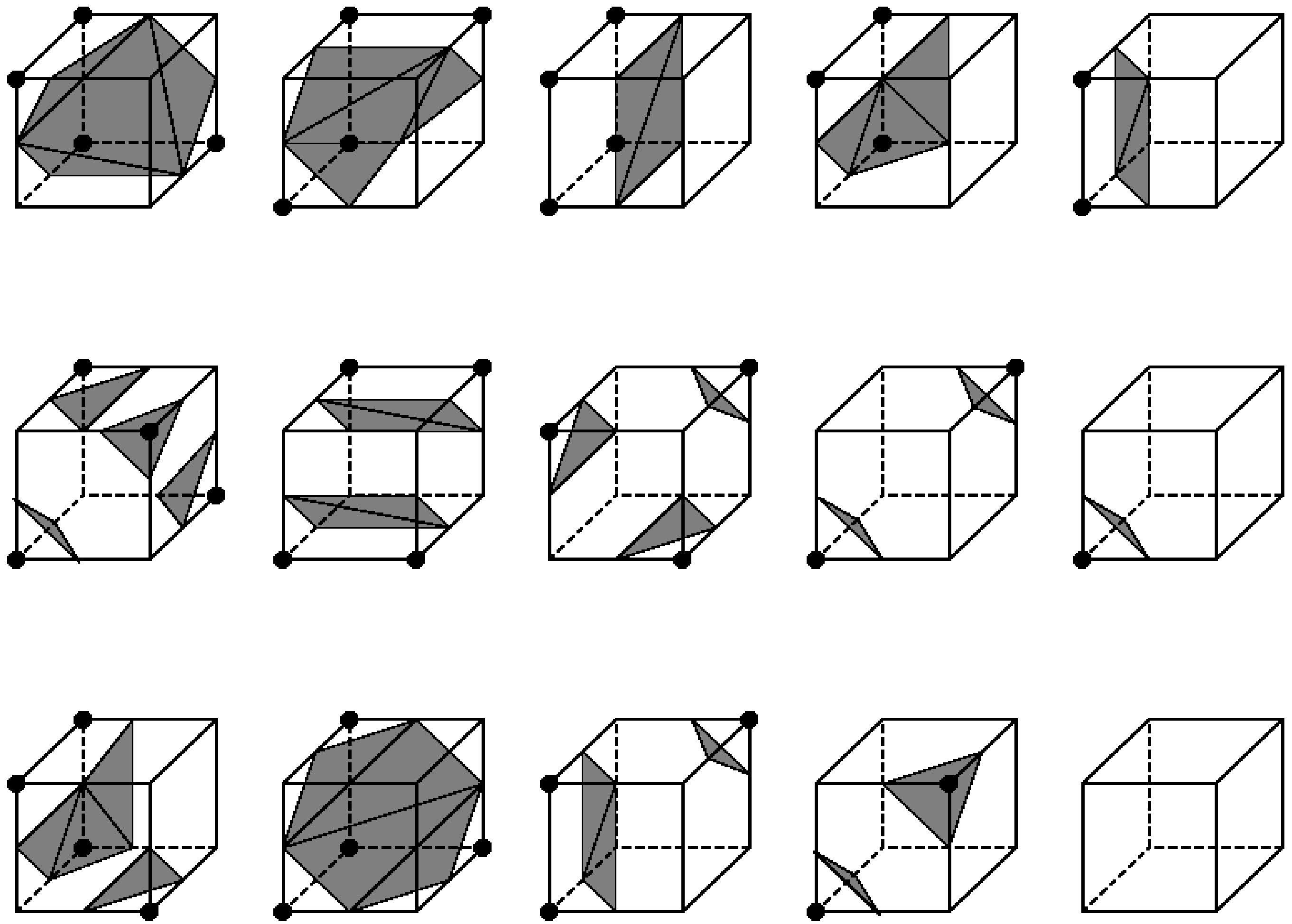


Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Lookup Mesh Connectivity



Cell Connectivity Table



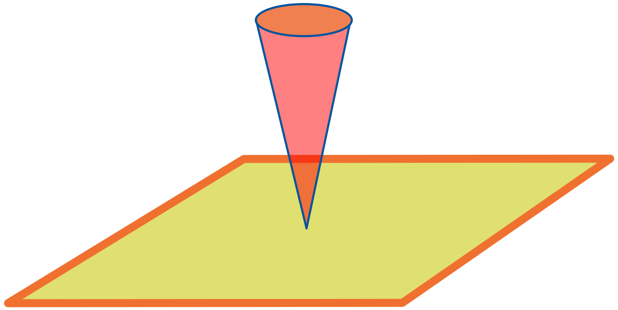
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Extended Marching Cubes

- same algorithmic principle

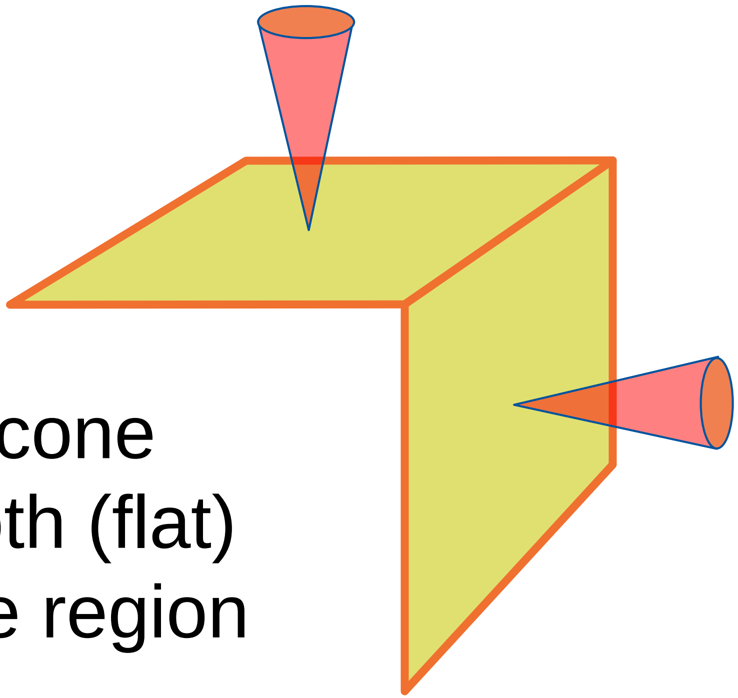
- for each cell ...
  - Find surface samples on the grid cell edges
  - Evaluate surface normals at the sample points
  - Feature detection
  - Feature sampling
  - Feature reconstruction

Feature Classification



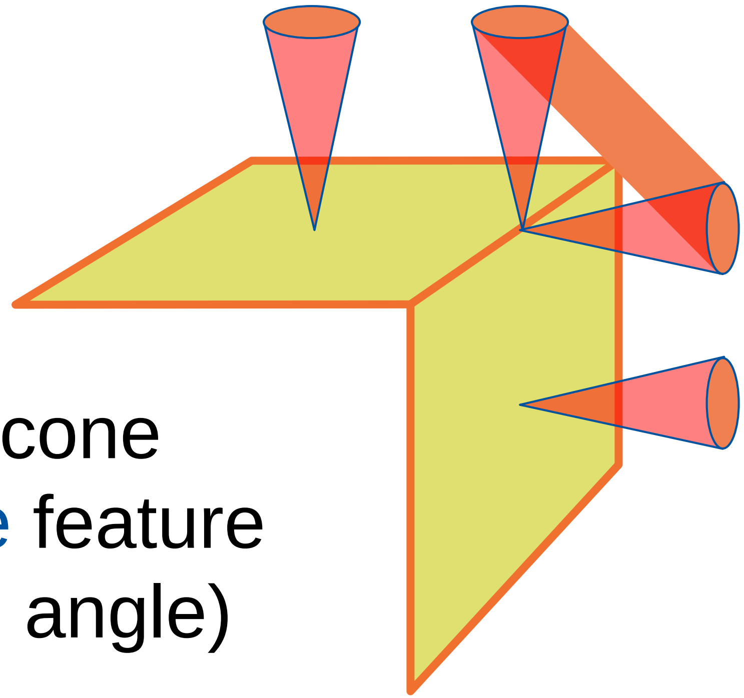
Normal cone
for a smooth (flat)
non-feature region

Feature Classification



Normal cone
for a smooth (flat)
non-feature region

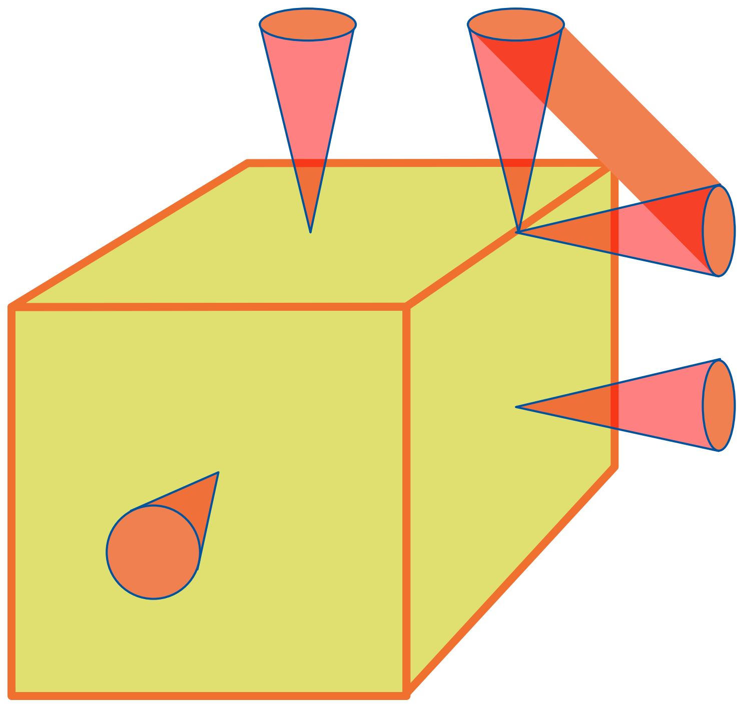
Feature Classification



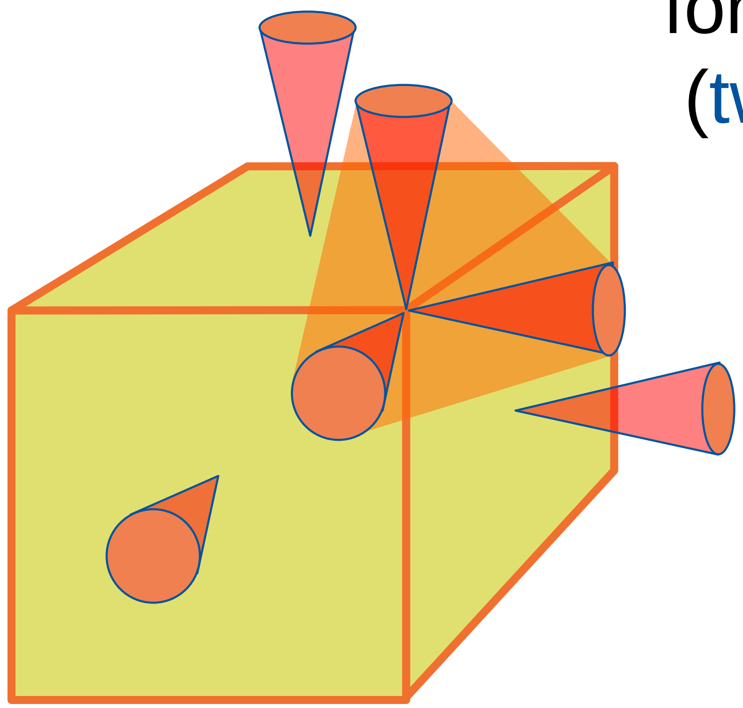
Normal cone
for an edge feature
(one large angle)

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Feature Classification



Feature Classification



Normal cone
for a corner feature
(two large angles)

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Feature Detection

- opening angles of the normal cone

  $\theta = \max_{i,j} \langle n_i, n_j \rangle$

  $\varphi = \min_k \langle n_k, n_i \times n_j \rangle$

- edge features

  $\theta > \theta_0, \ \ \varphi > \varphi_0$ → sharp *edge*

- corner features

  $\theta > \theta_0, \ \ \varphi <= \varphi_0$ → sharp *corner*

Feature Sampling

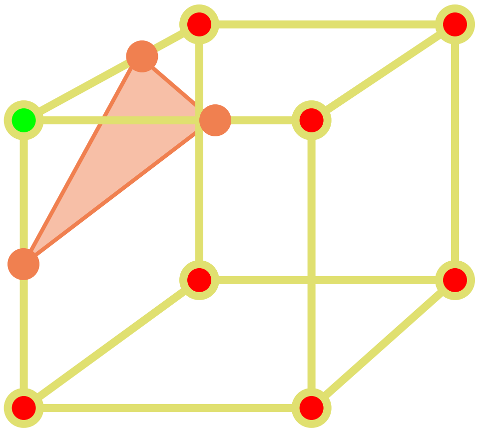
- Each pair $(\boldsymbol{p}_\mathrm{i}, \boldsymbol{n}_\mathrm{i})$ defines a ***tangent element***

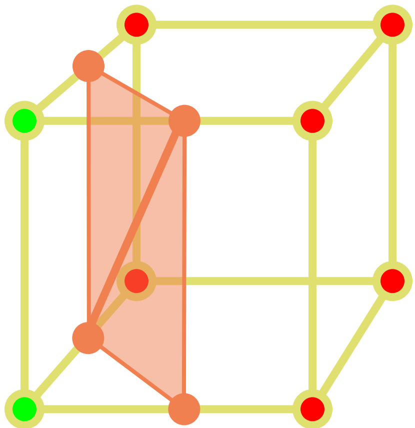
- Intersect tangent planes to approximate a ***piecewise*** smooth surface

- Find least squares solution by **SVD**
  - Overdetermined cases
  - Underdetermined cases
  - Suppress smallest singular value at edge features

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Lookup Mesh Connectivity

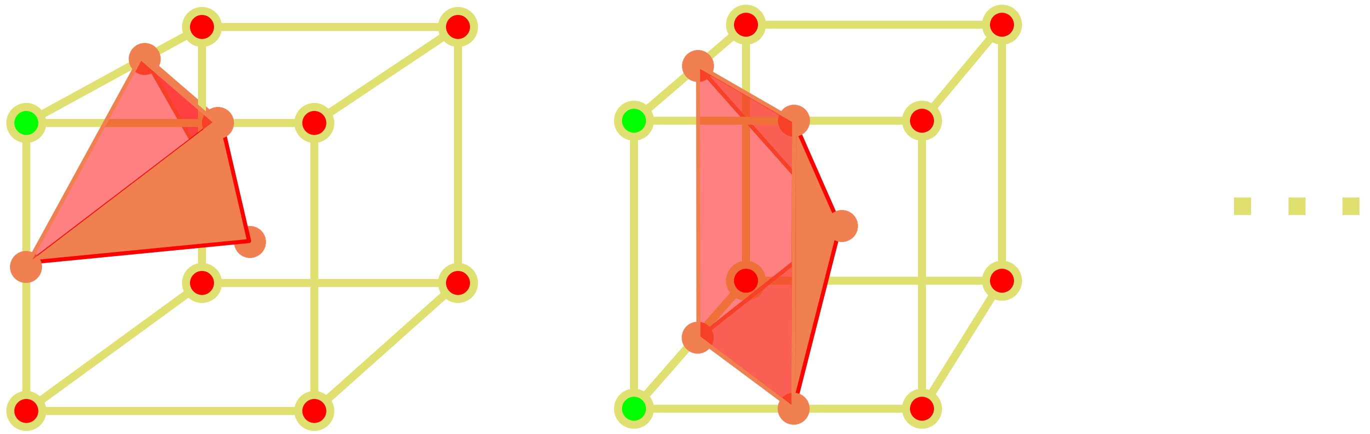




. . .

Modified Lookup Table

- Generate triangle fans centered around the feature sample



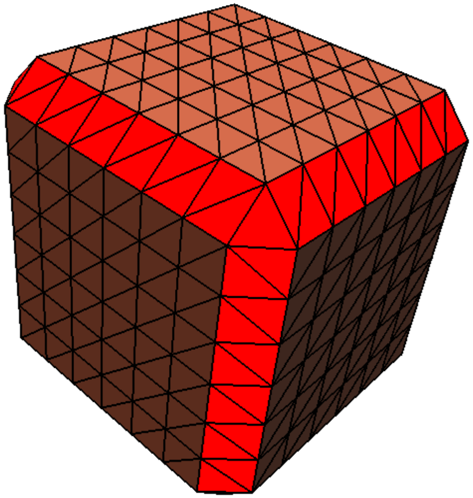
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

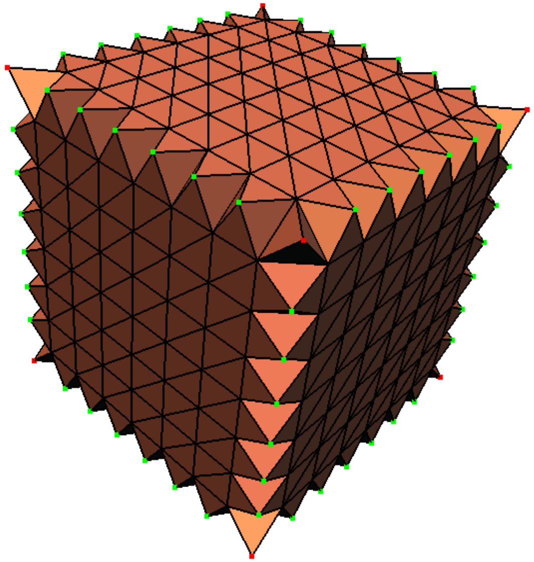
Feature Reconstruction

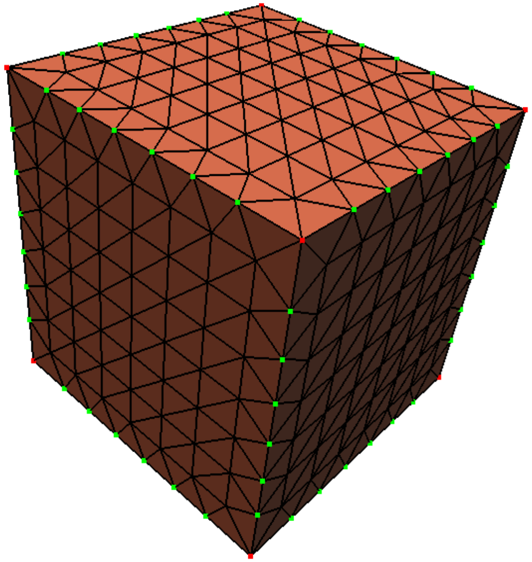
feature detection

feature sampling

edge-flipping







Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Results

- Mesh complexity
  - Two additional triangles per feature sample
  - Typically 10% overhead
- Computation time
  - 20% to 40% overhead
- Improved approximation
- Parameters θ_0 and φ_0
  - False positive feature detection !?
- Applications ...

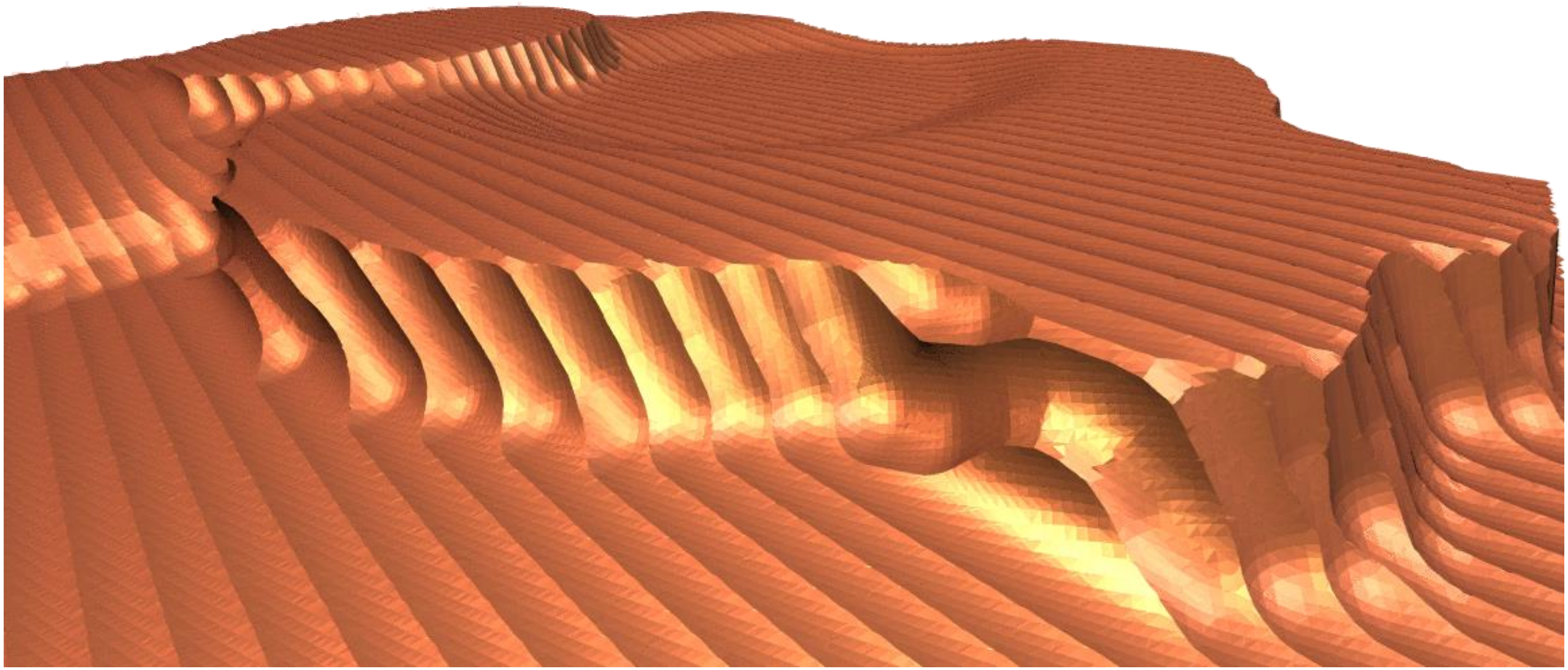
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

CSG ... CAD / CAM

65 x 65 x 65

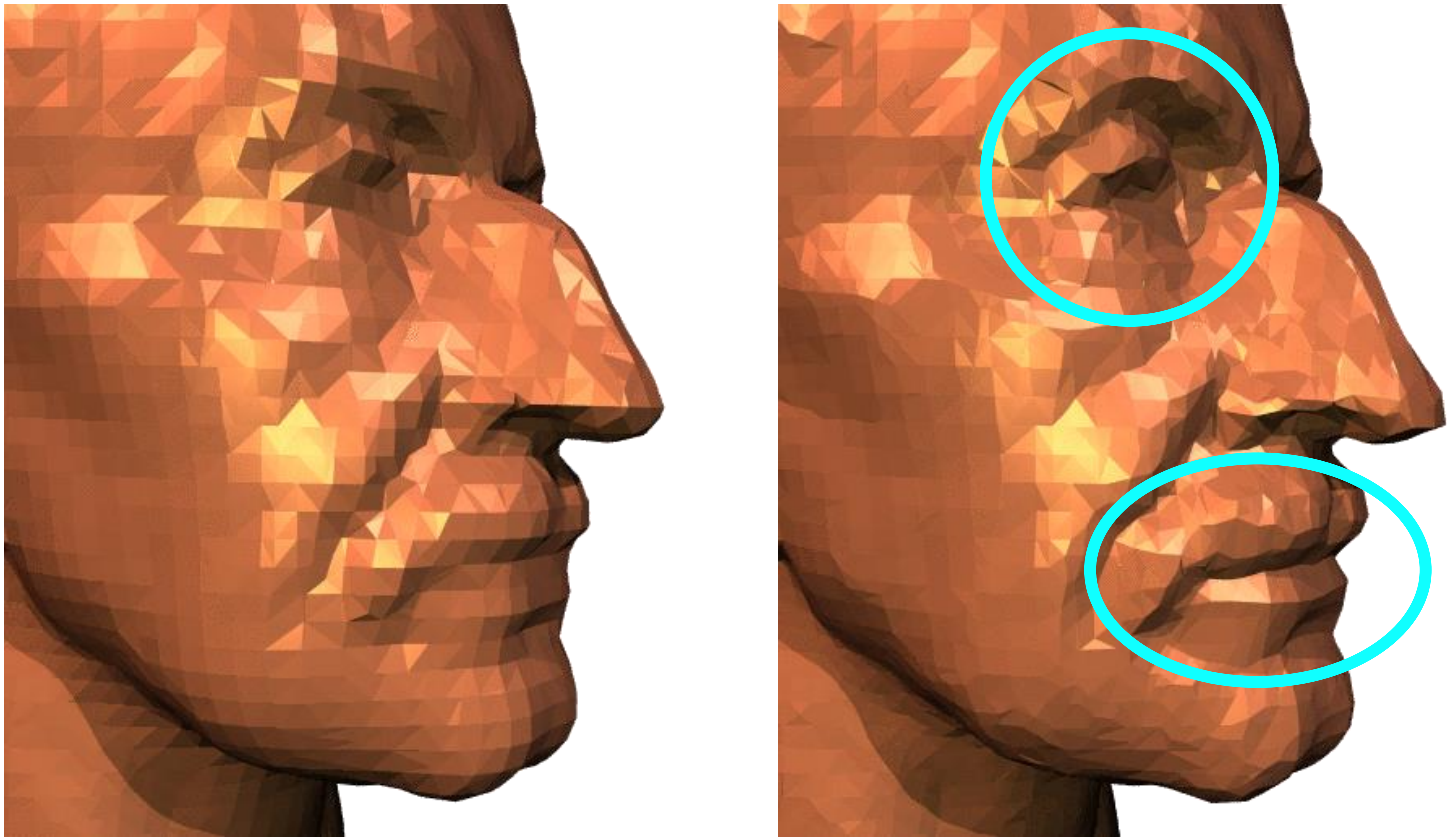
CSG ... Milling Simulation

257 x 257 x 257



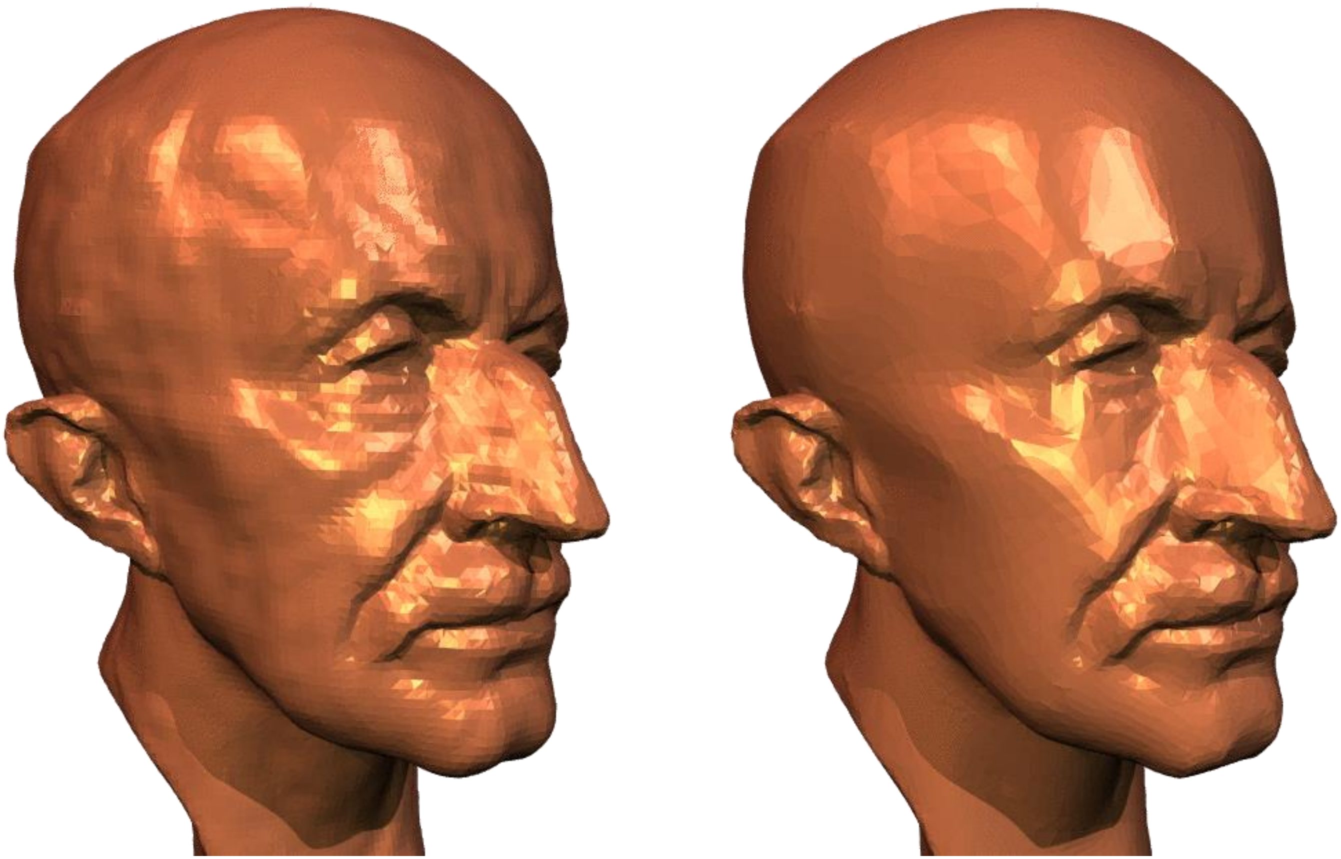
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Iso-Surface Extraction



65 x 65 x 65

Iso-Surface Extraction



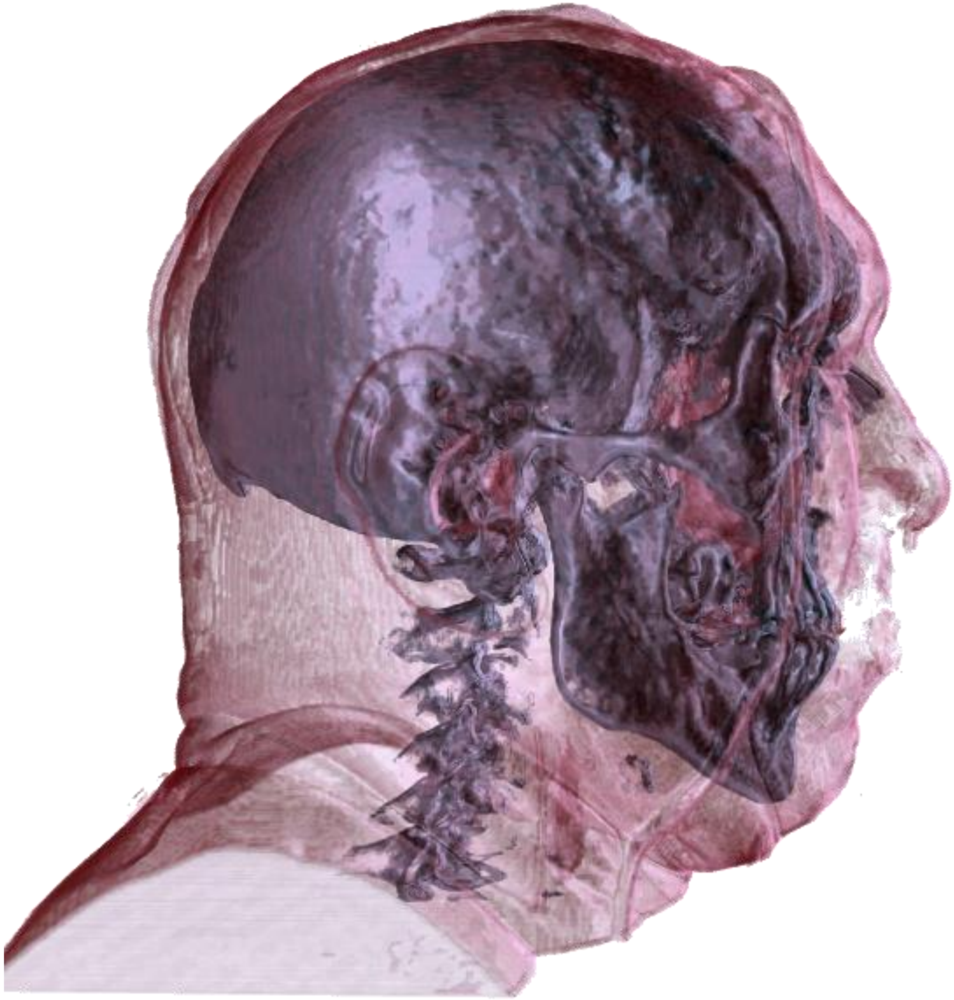
Volumetric Representations

- *Indirect Rendering*:
  - Isosurface extraction: *Marching Cubes*

- *Direct Rendering*:
  - Ray Casting

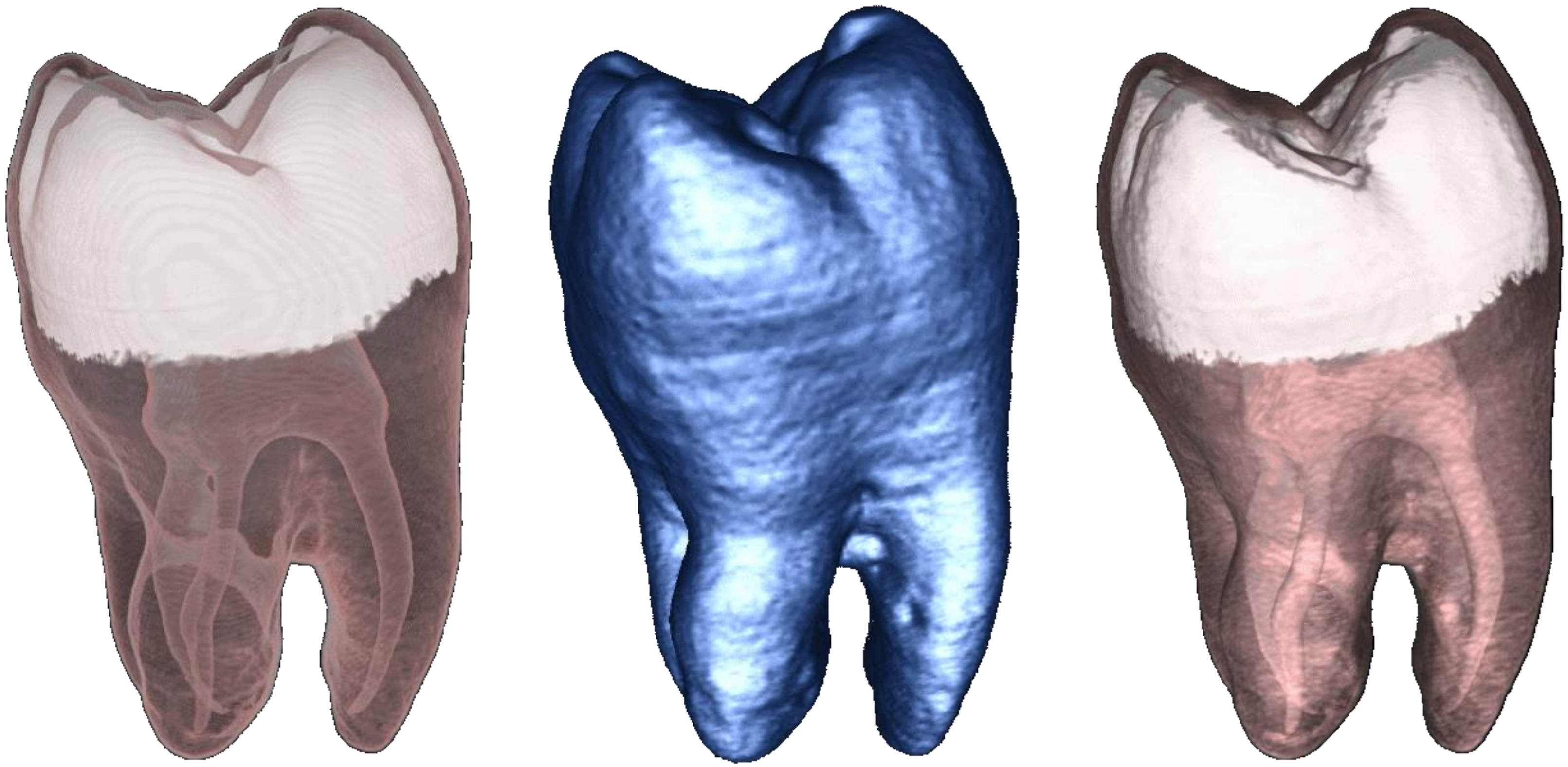
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volume Rendering

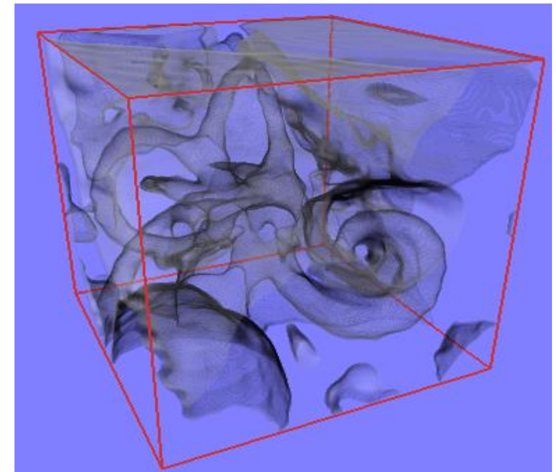


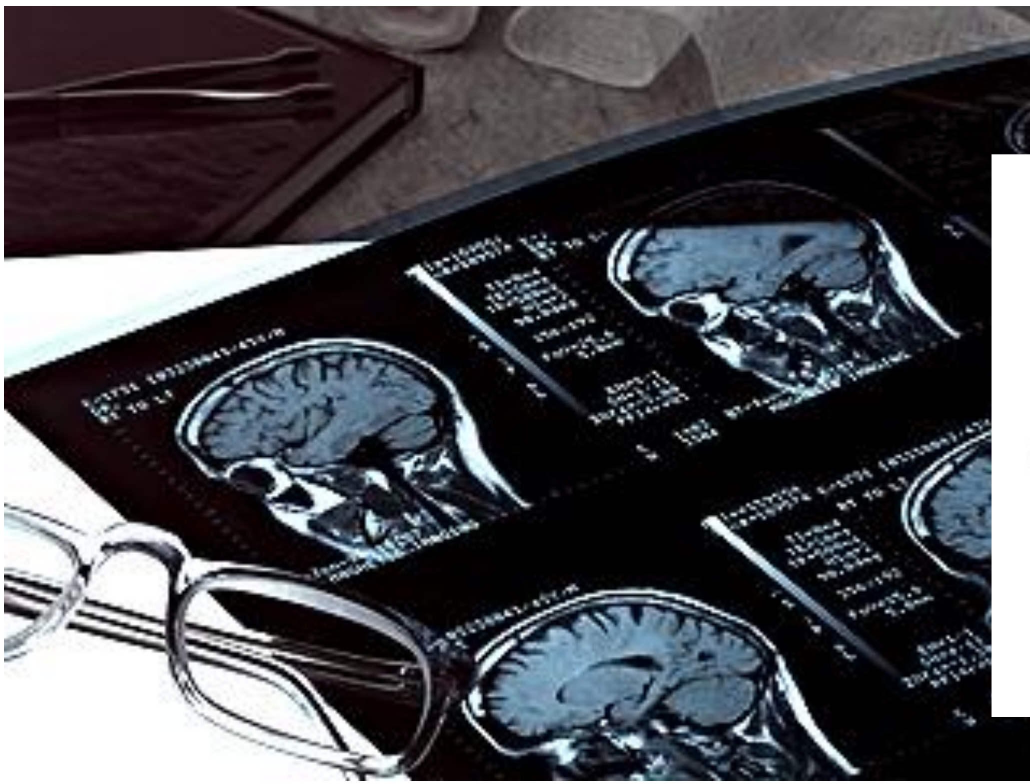
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volume Rendering

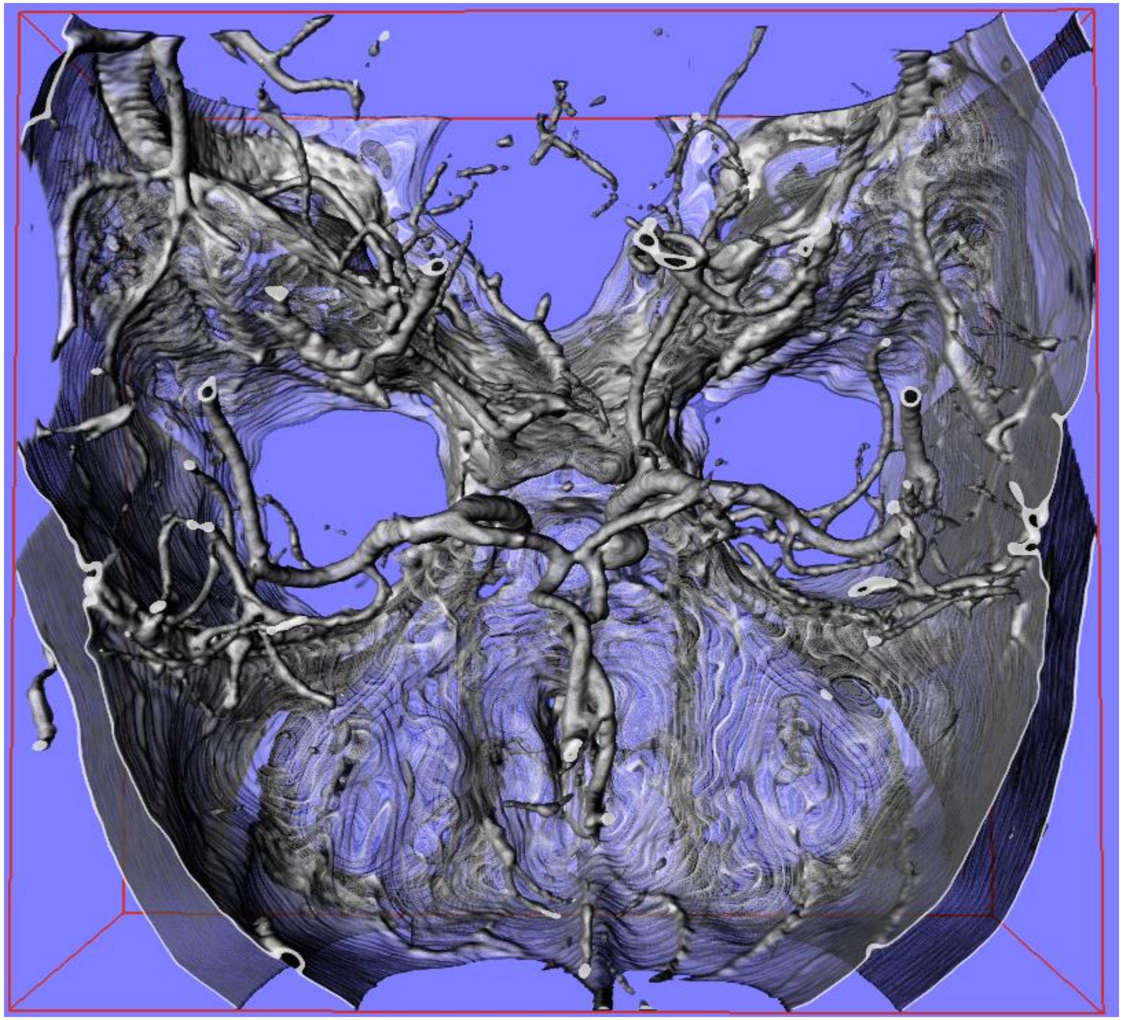


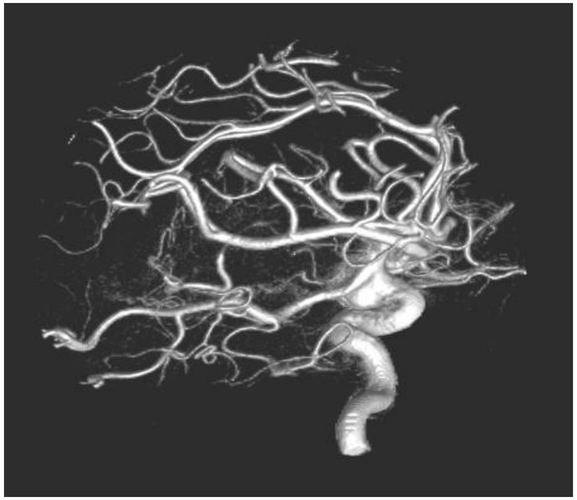
Volume Rendering





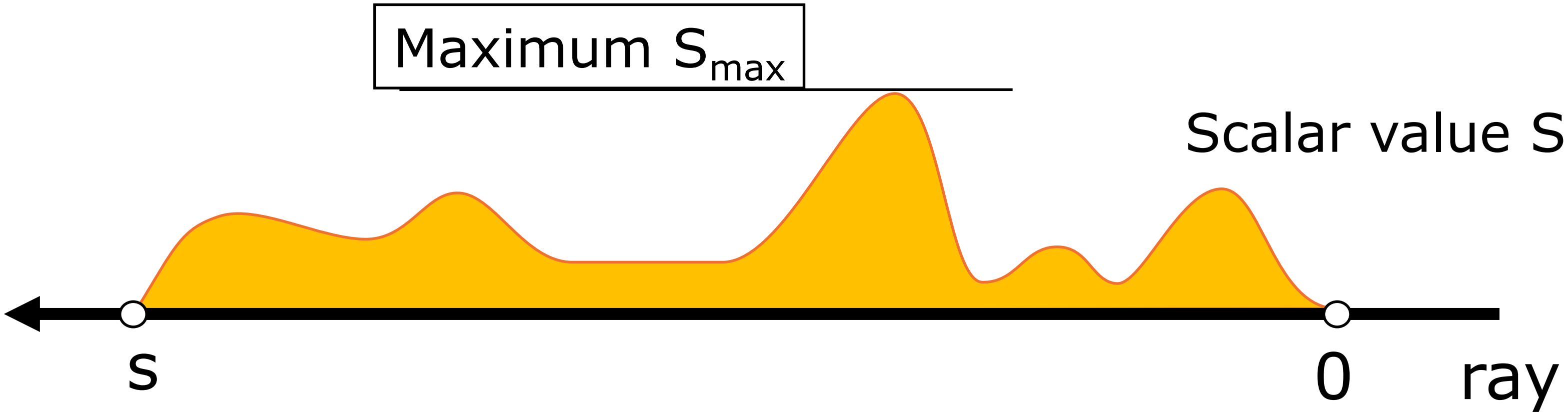
Volume Rendering



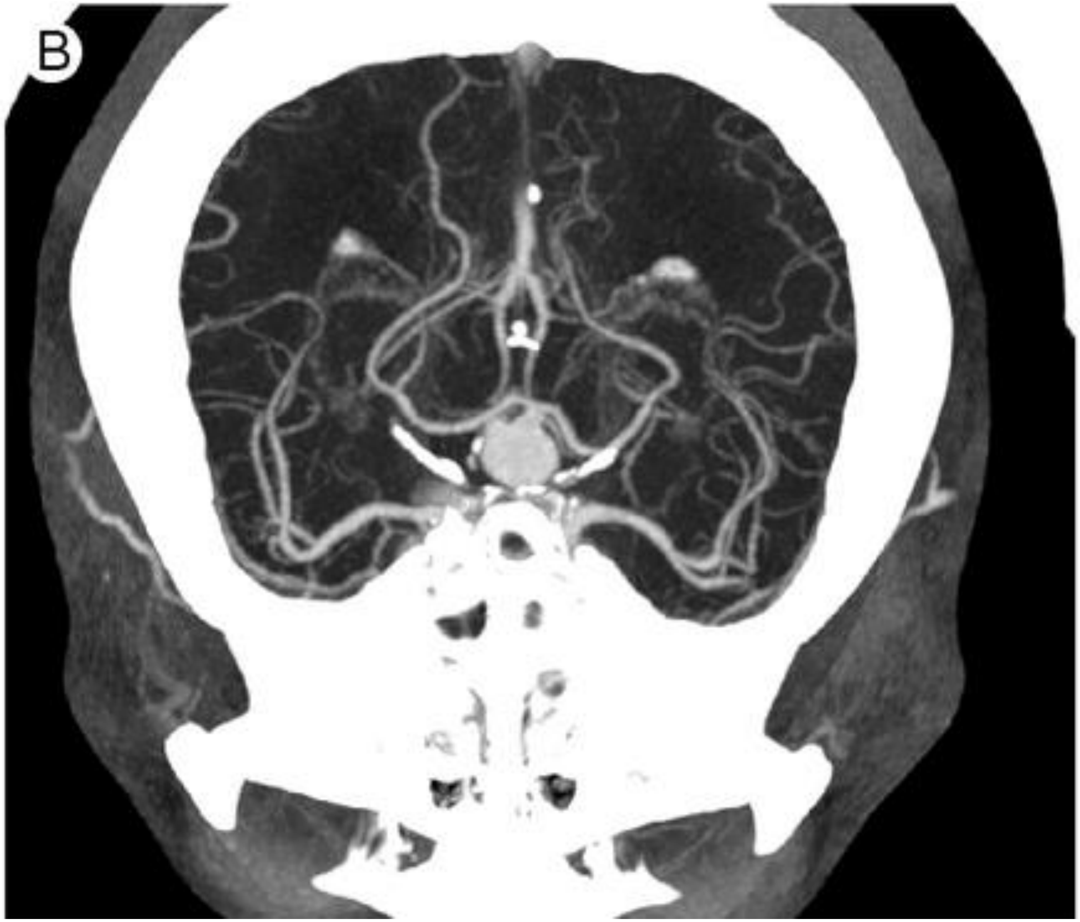


Ray Casting

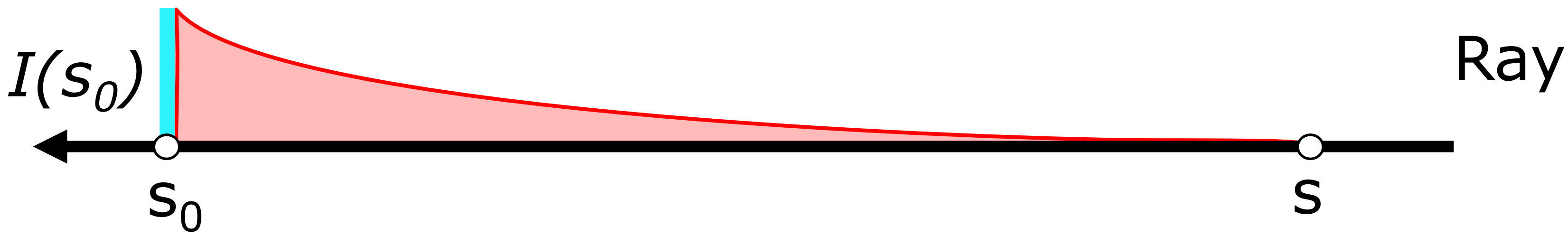
- Maximum Intensity Projection



Maximum Intensity Projection

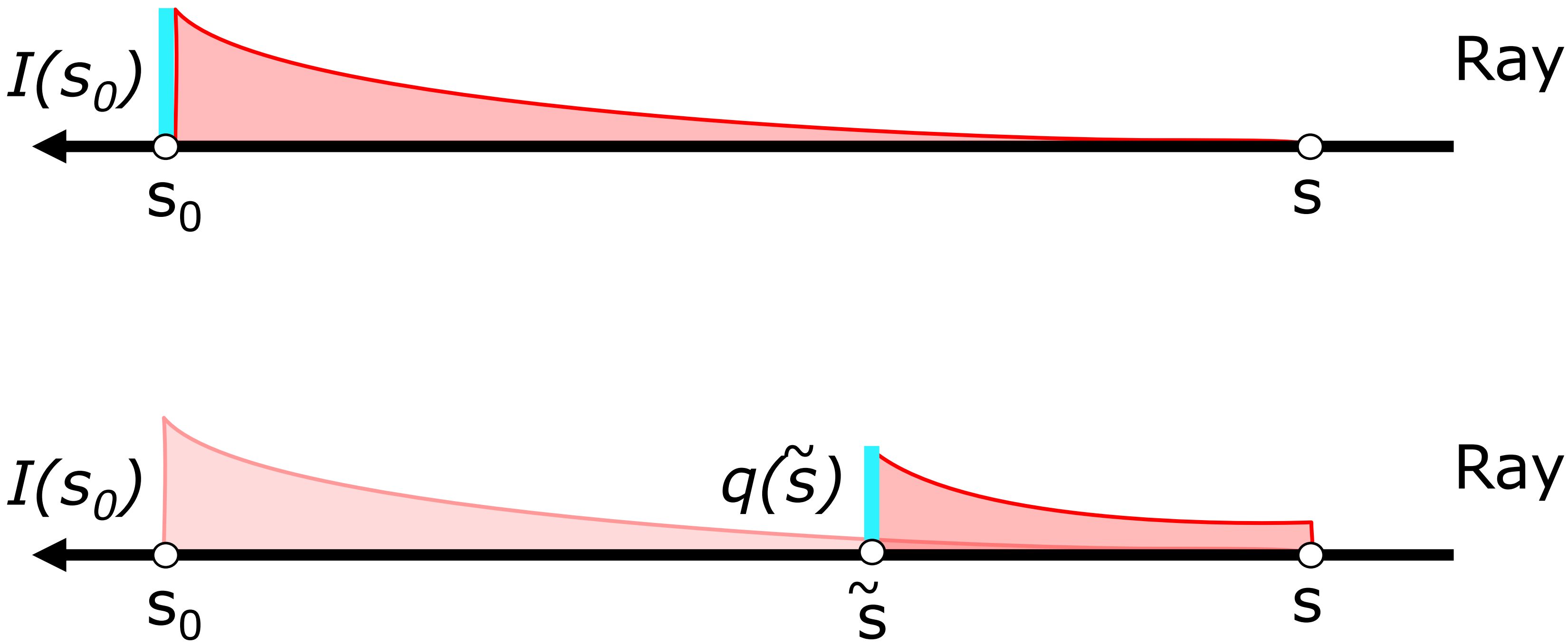


Absorption



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Absorption



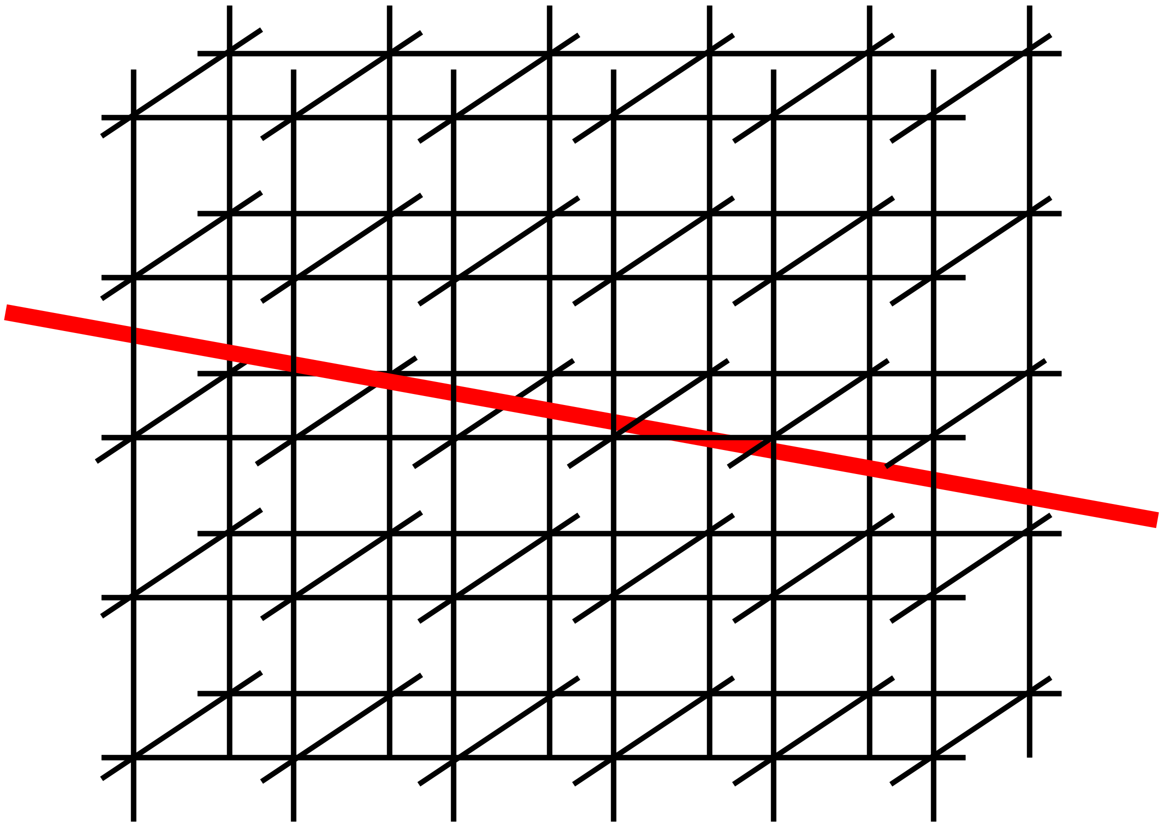
Ray Casting

- *Image order*: for each pixel ...
- Shoot rays from eye through each image plane pixel
- Integrate color c & opacity μ along the ray r

$$c(r) = \int_0^L c(s) \cdot e^{-\int_0^s \mu(t)\,dt}\, ds$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volumetric Representations



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Opacity

- base color c, opacity μ

  → damping factor $1-\alpha = e^{-\mu}$
  → emitted color $\alpha\, c$

- resulting (effective) color

  → $\alpha\, c + (1-\alpha)\, c_{background}$

Ray Casting

set color $c(.)$ and opacity $\alpha(.)$ for all voxels

for all pixels $\boldsymbol{u}_i = (u_i, v_i)$

find ray from eye through pixel $\boldsymbol{u}_i$: $eye + \lambda \boldsymbol{d}$

init $c_{acc}(\boldsymbol{u}_i) = \alpha_{acc}(\boldsymbol{u}_i) = 0$

for all samples $\boldsymbol{x}_j = (x_j, y_j, z_j) = eye + \lambda_j \boldsymbol{d}$

get $c(\boldsymbol{x}_j)$ and $\alpha(\boldsymbol{x}_j)$ by tri-linear interpolation

composite with $c_{acc}(\boldsymbol{u}_i)$ and $\alpha_{acc}(\boldsymbol{u}_i)$

Opacity Accumulation

- **Continous** compositing

$$c(u_i) = \int_0^L c(\mathbf{E} + \lambda \mathbf{D}) \cdot e^{-\int_0^\lambda \mu(\mathbf{E}+t\mathbf{D})\,dt} \, d\lambda$$

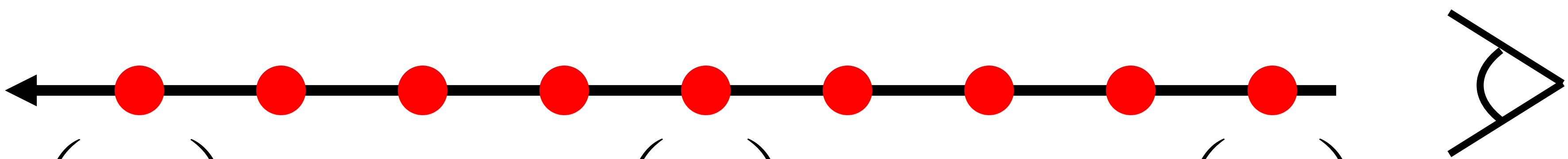
- **Discrete** compositing

$$c(u_i) = \sum_{j=0}^{K} \left[c(x_j) \cdot \alpha(x_j) \cdot \prod_{m=0}^{j-1} (1 - \alpha(x_m)) \right]$$

$$\text{with} \quad c(x_k) = c_{bkg}, \quad \alpha(x_k) = 1$$

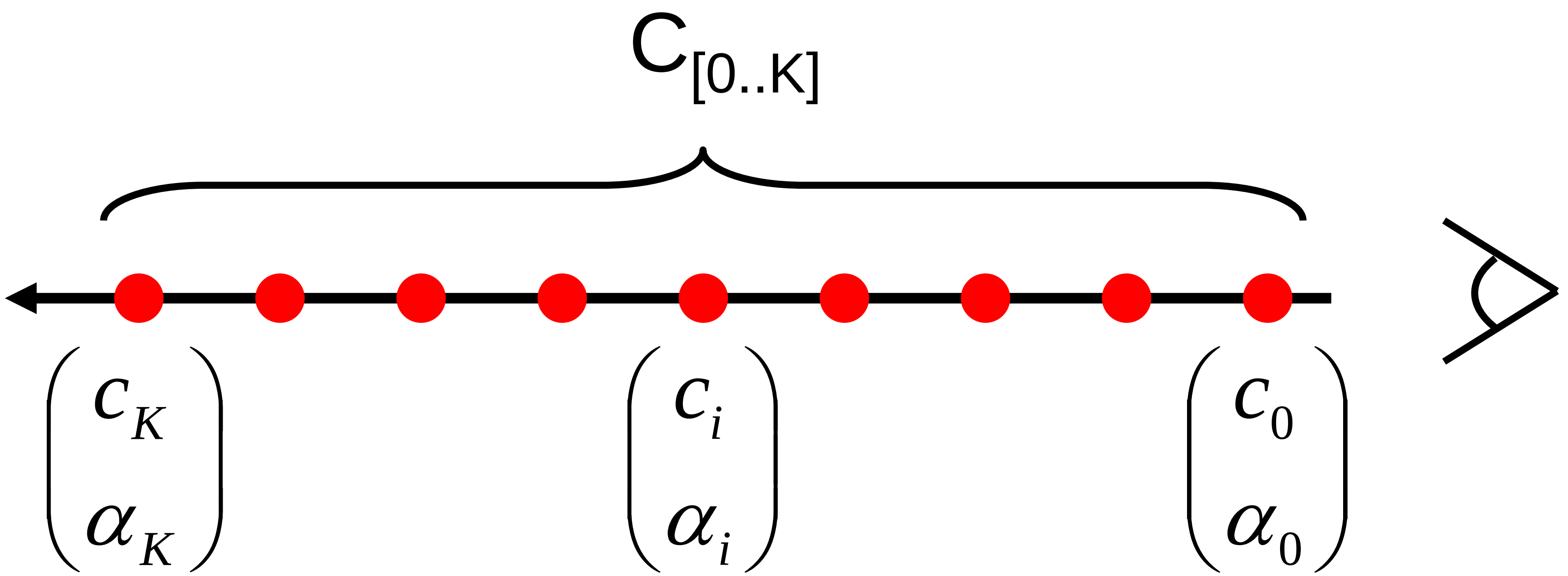
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Opacity Accumulation



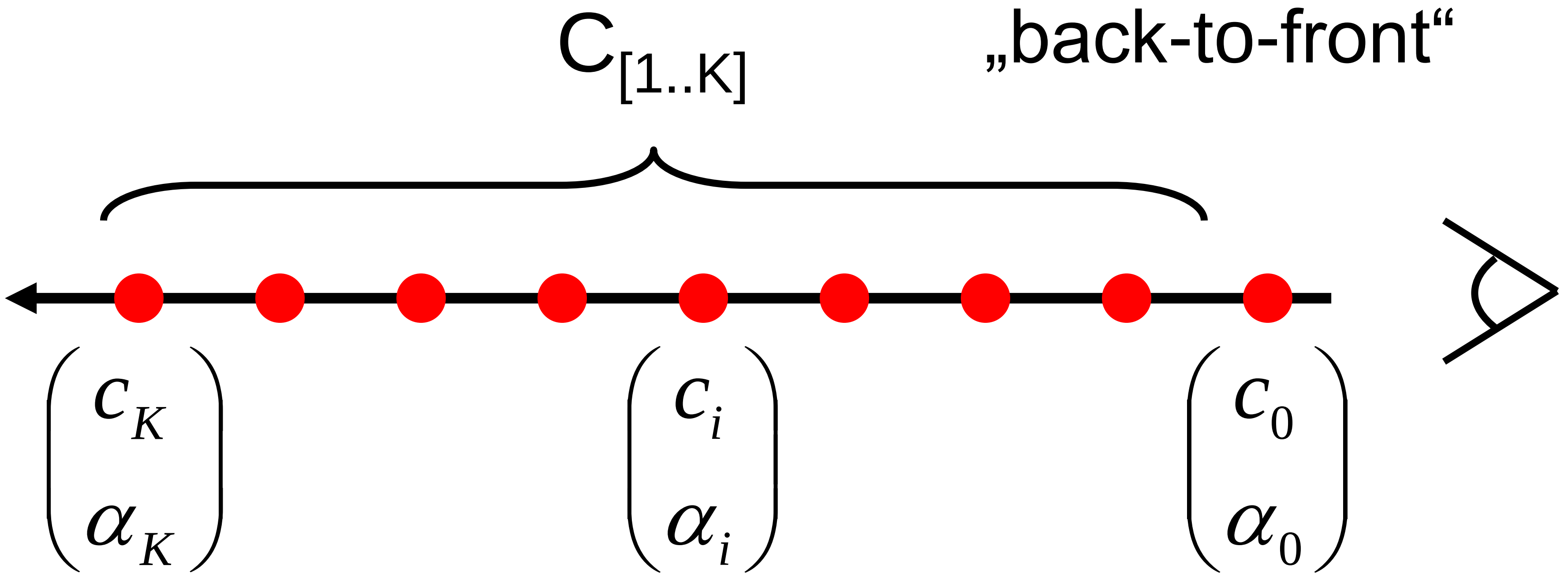
$$\begin{pmatrix} c_K \\ \alpha_K \end{pmatrix} \qquad \begin{pmatrix} c_i \\ \alpha_i \end{pmatrix} \qquad \begin{pmatrix} c_0 \\ \alpha_0 \end{pmatrix}$$

Opacity Accumulation

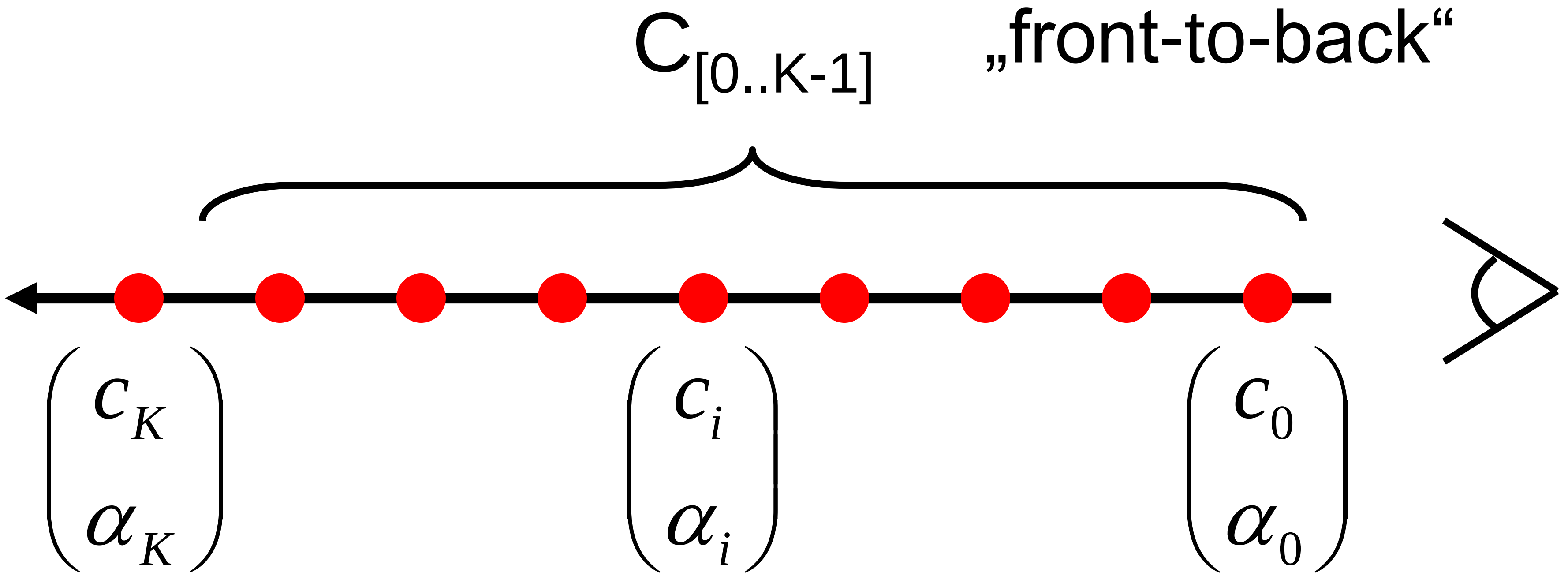


Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Opacity Accumulation



Opacity Accumulation



Back-to-Front Compositing

$$\sum_{j=0}^{K}\left[c(x_j)\cdot\alpha(x_j)\cdot\prod_{m=0}^{j-1}(1-\alpha(x_m))\right]$$

$$=c(x_0)\cdot\alpha(x_0)+$$

$$\sum_{j=1}^{K}\left[c(x_j)\cdot\alpha(x_j)\cdot\prod_{m=1}^{j-1}(1-\alpha(x_m))\right]\cdot(1-\alpha(x_0))$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Front-to-Back Compositing

- Iterative accumulation j = 0 .. K

$$C \leftarrow C + c_j \alpha_j (1 - \hat{\alpha})$$

$$\hat{\alpha} \leftarrow \hat{\alpha} + \quad \alpha_j (1 - \hat{\alpha})$$

with

$$(1 - \hat{\alpha}) = \prod_{m=0}^{j-1} (1 - \alpha_m)$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Transfer Functions

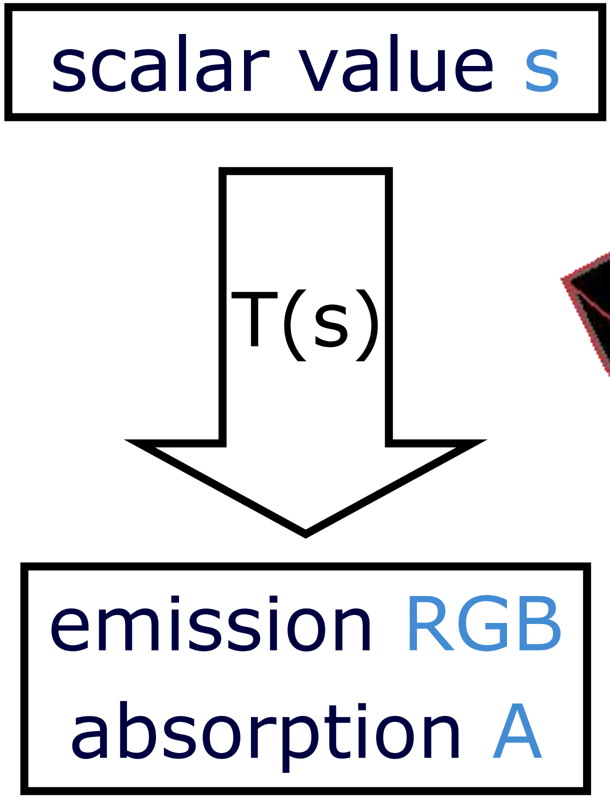
- how to define the
  - base color c
  - opacity α
- for each voxel ?

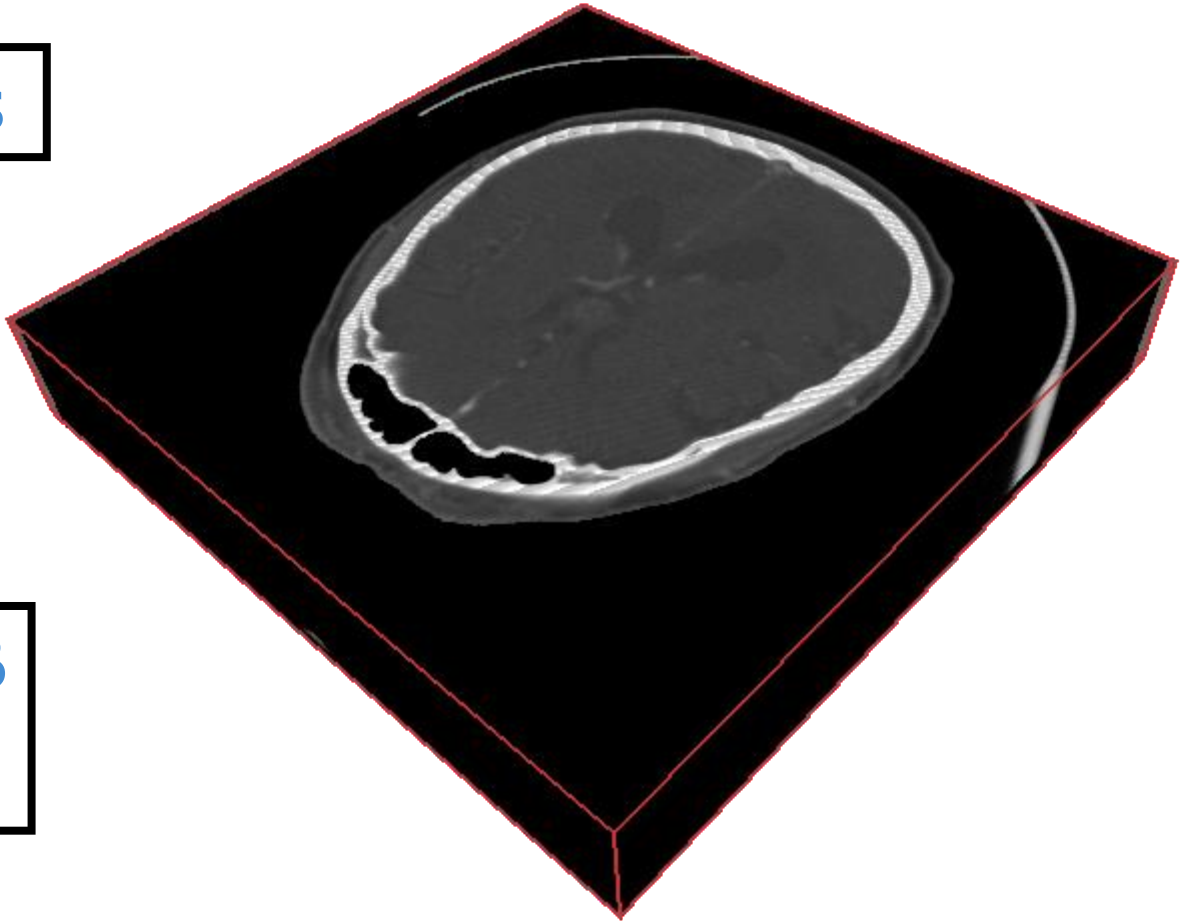
- CT, MRI, ...

- diffusion tensor imaging

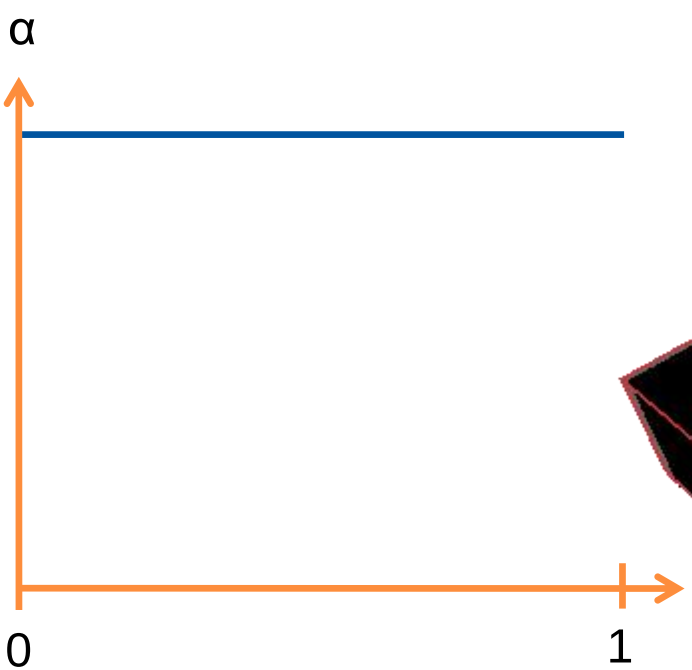
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

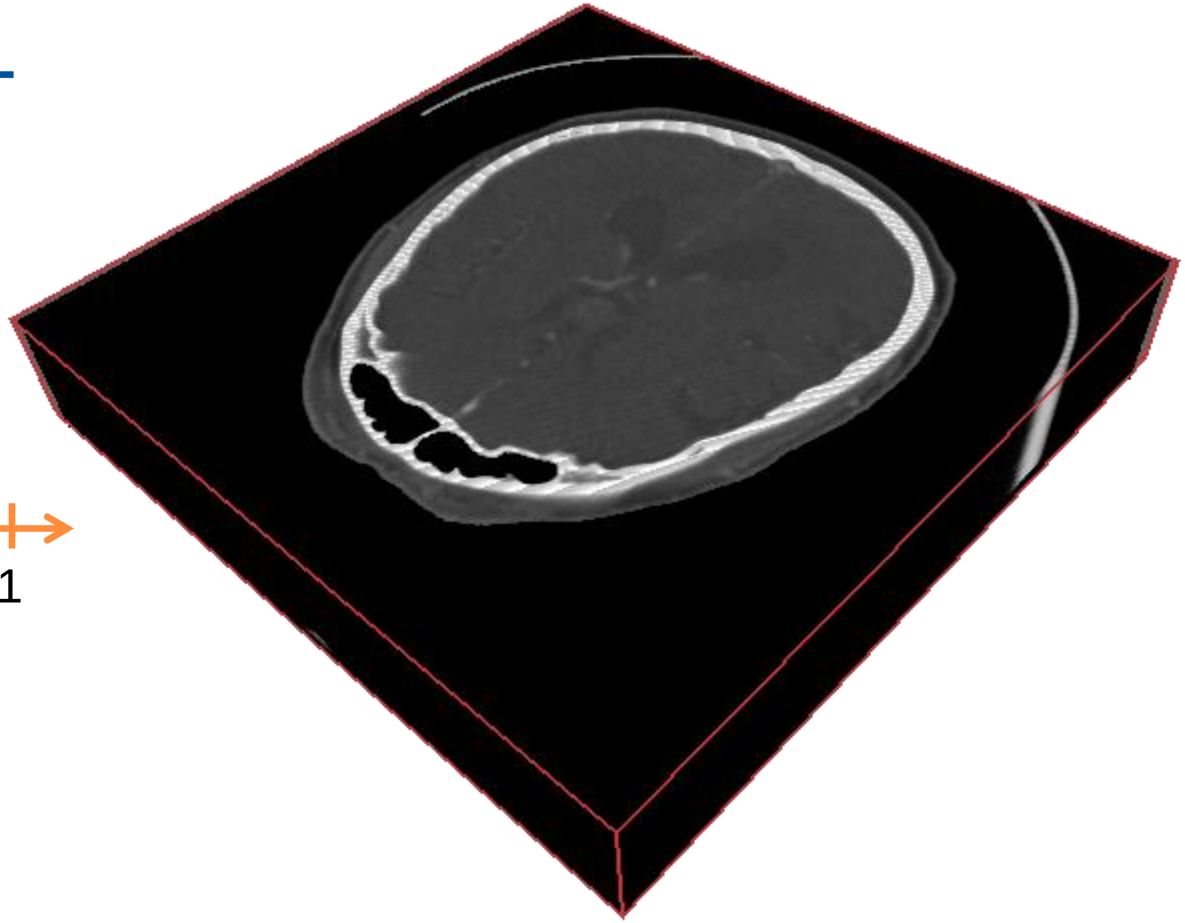
Transfer Functions





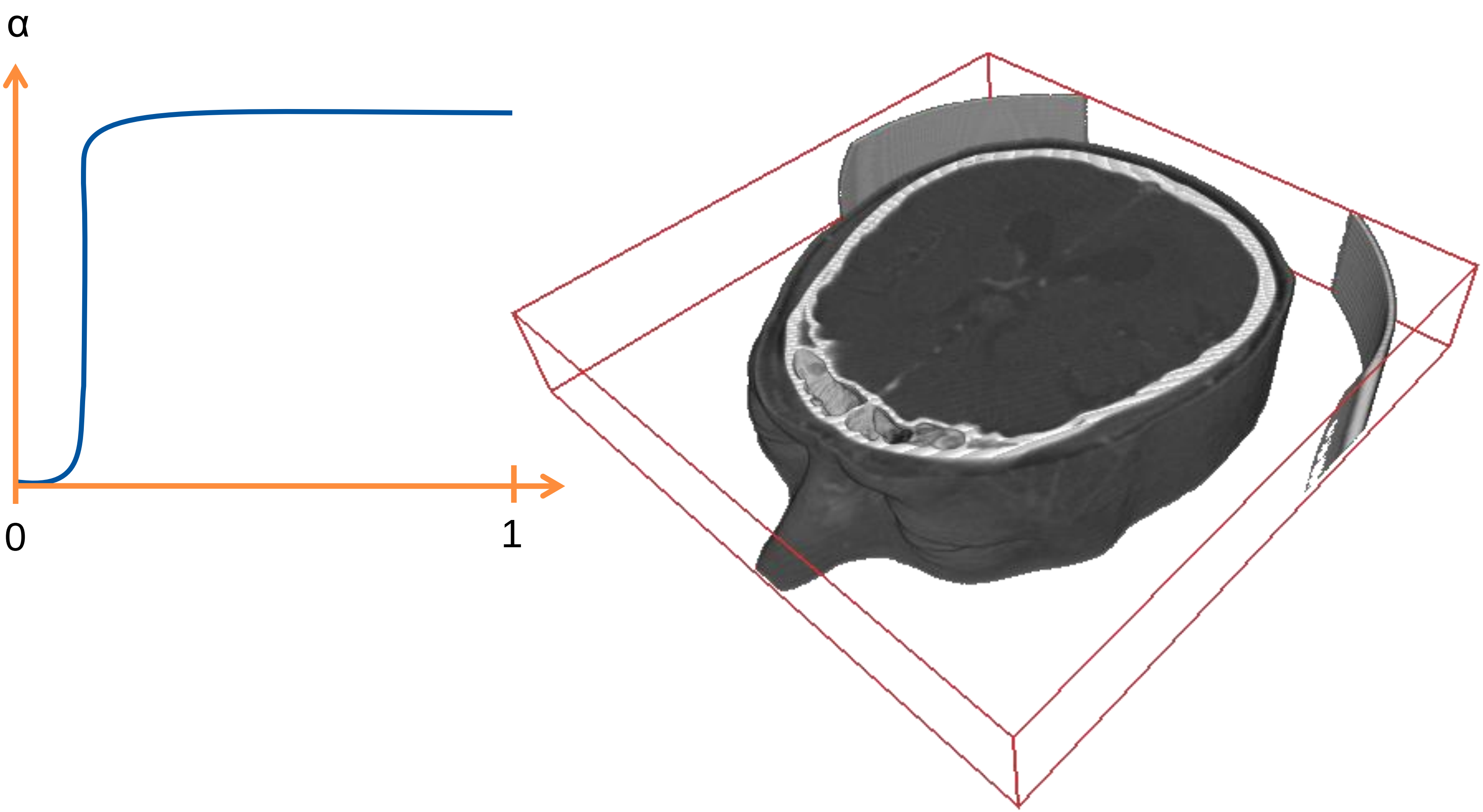
Transfer Functions



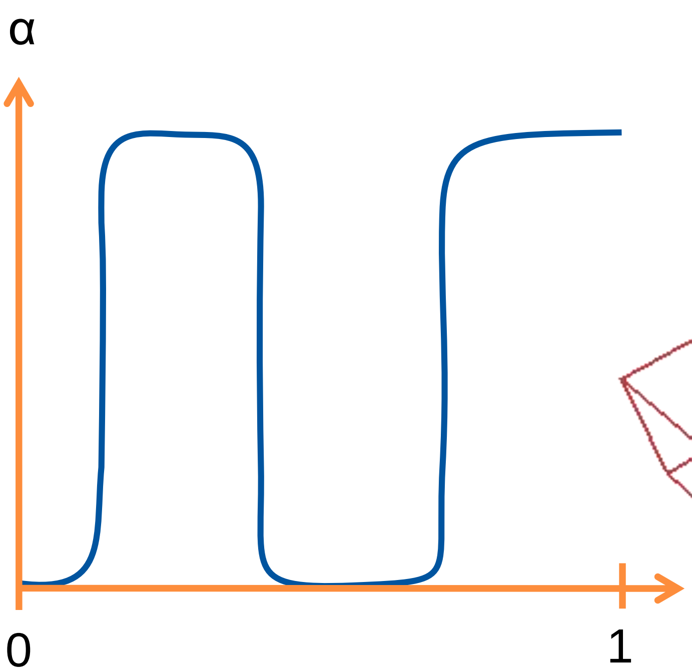


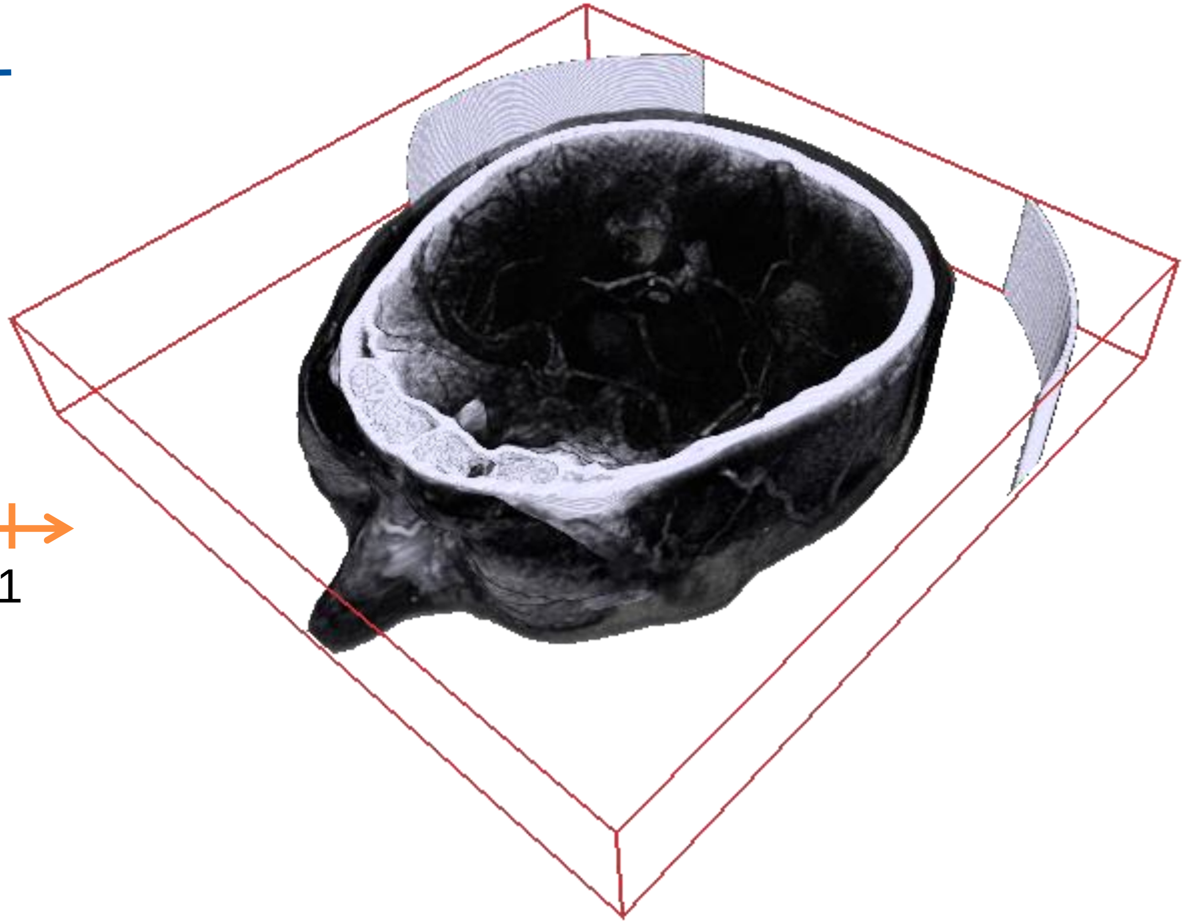
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Transfer Functions

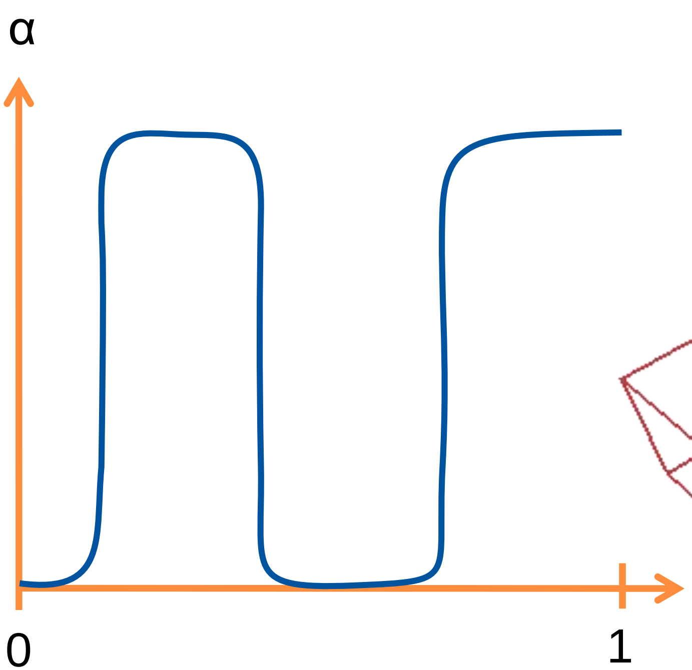


Transfer Functions

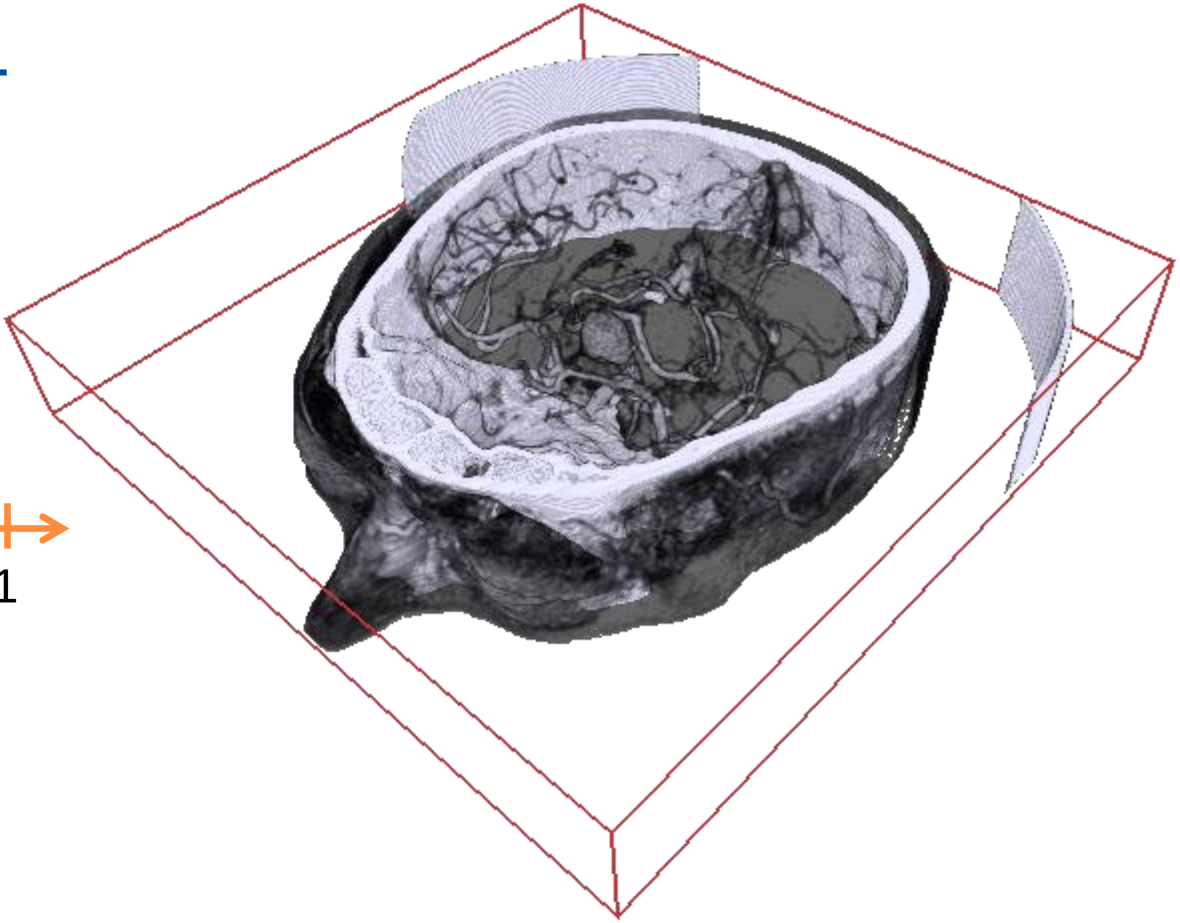




Transfer Functions

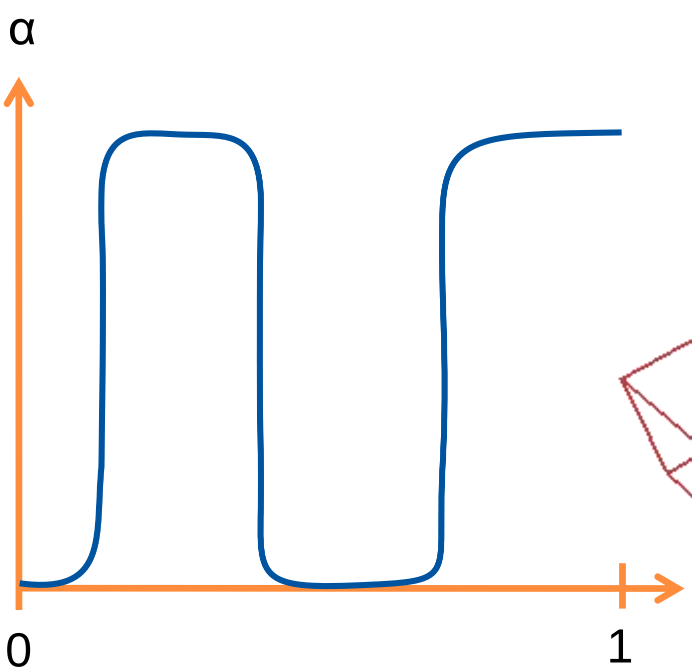


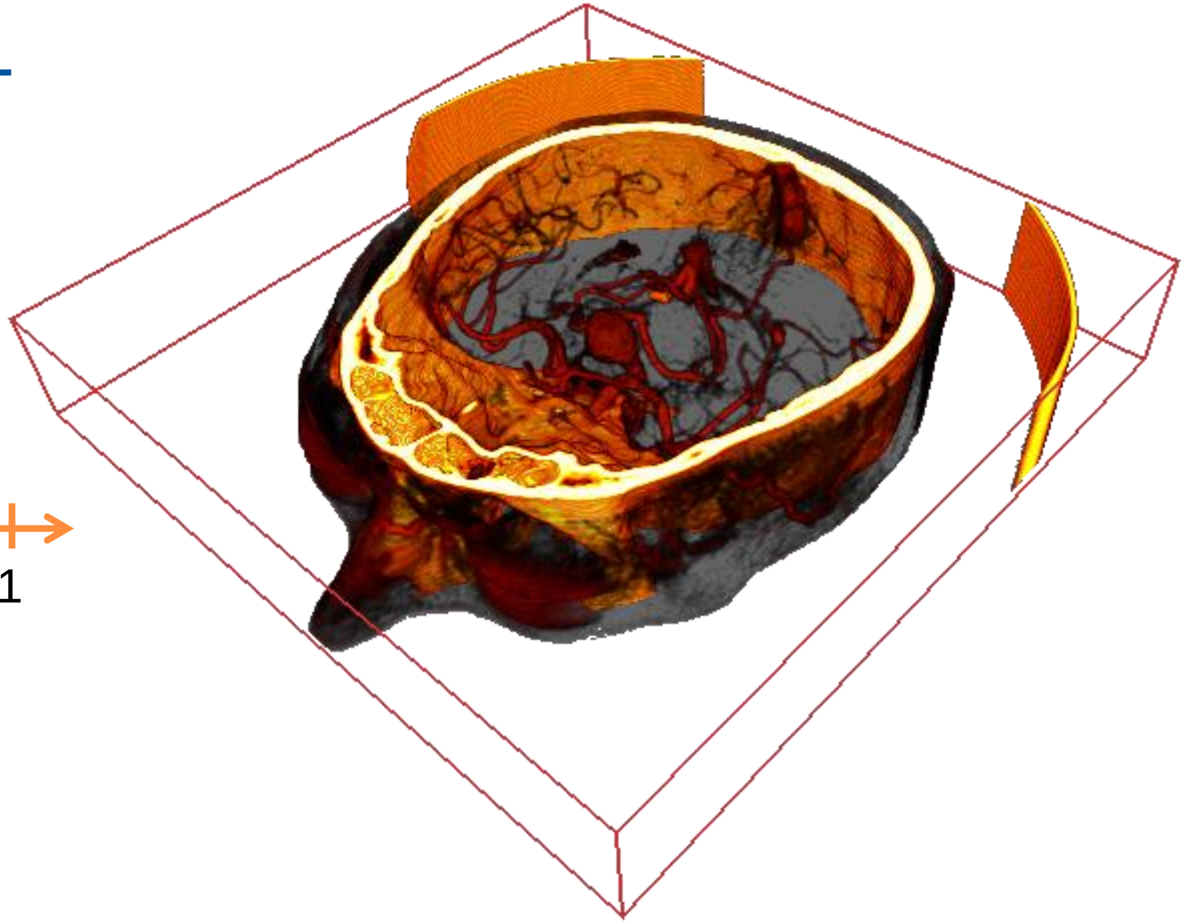
+ Shading



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

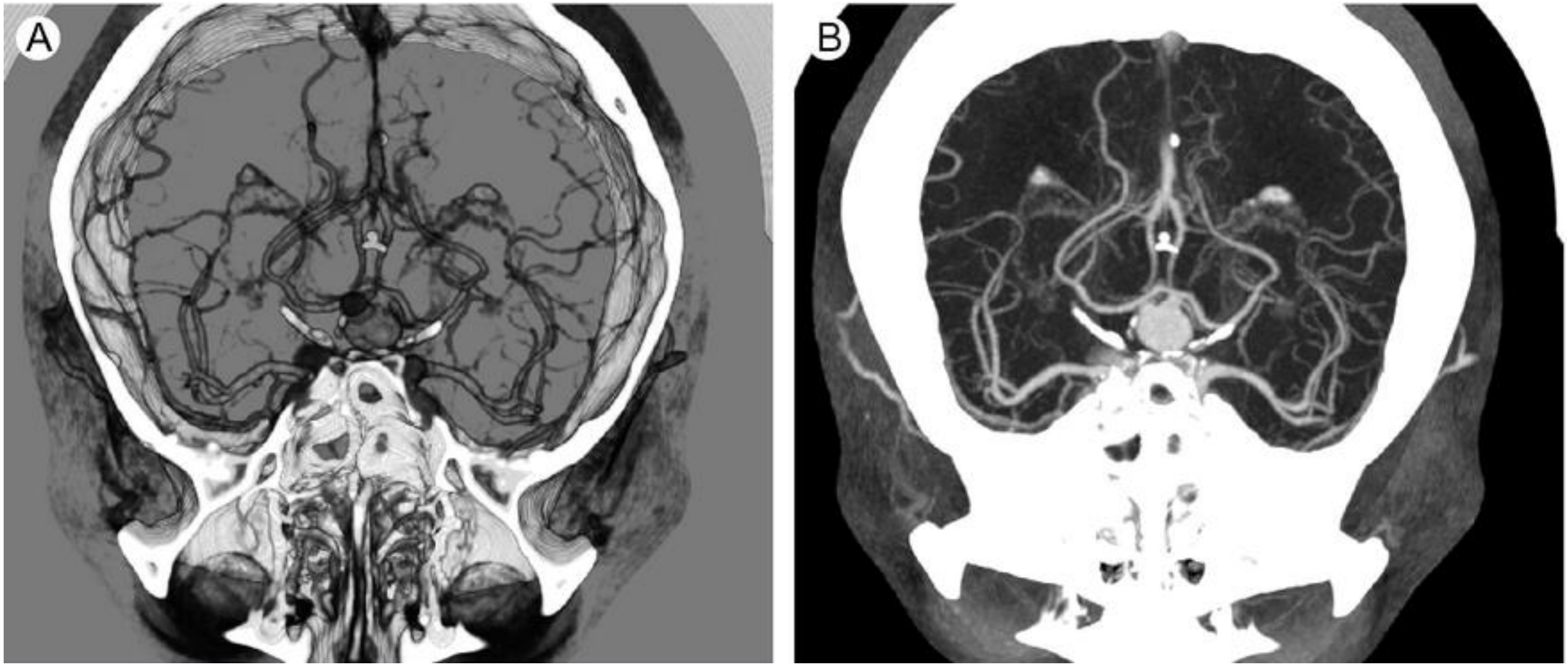
Transfer Functions



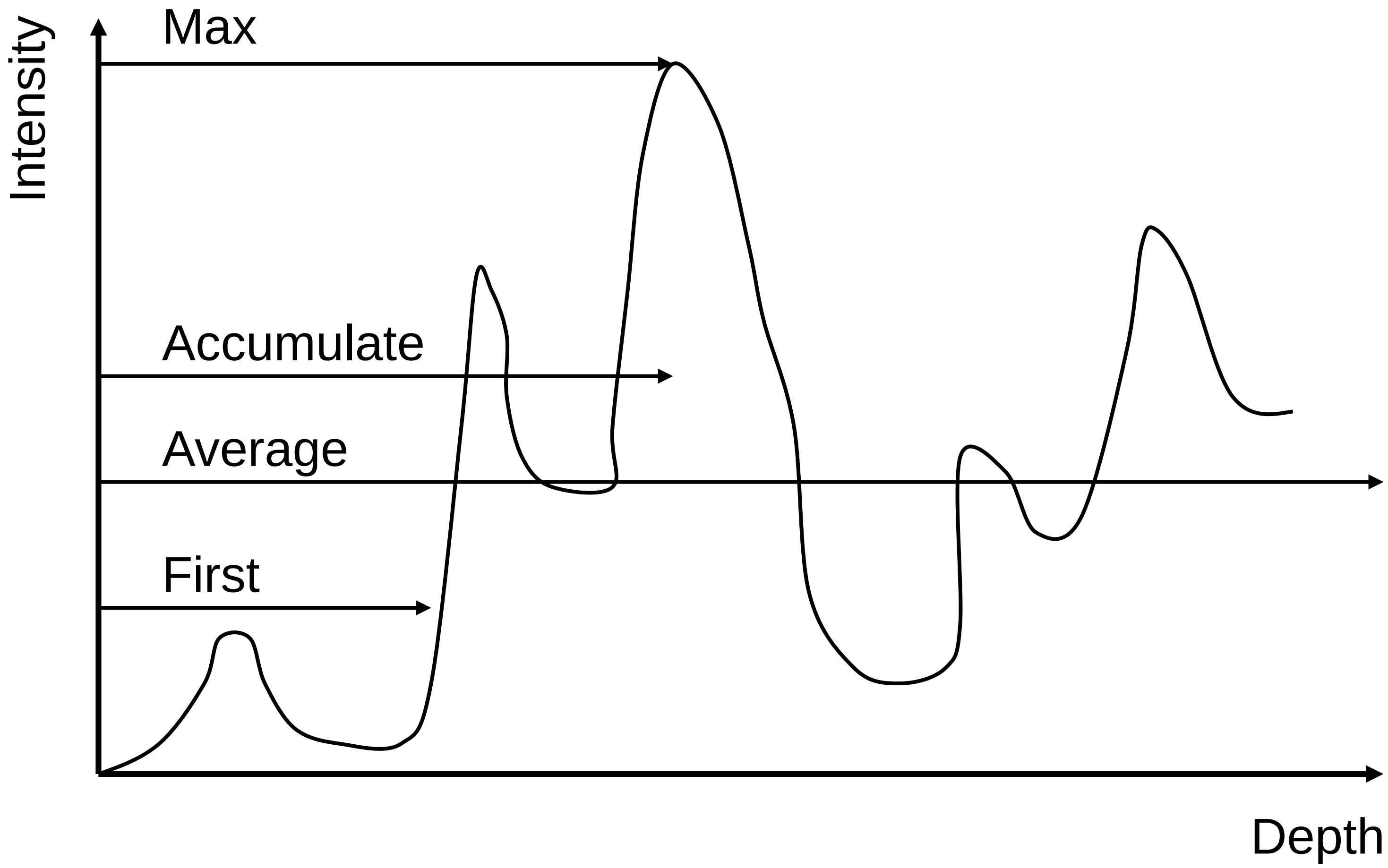


+ Shading
+ Color coding

Absorption vs. MIP

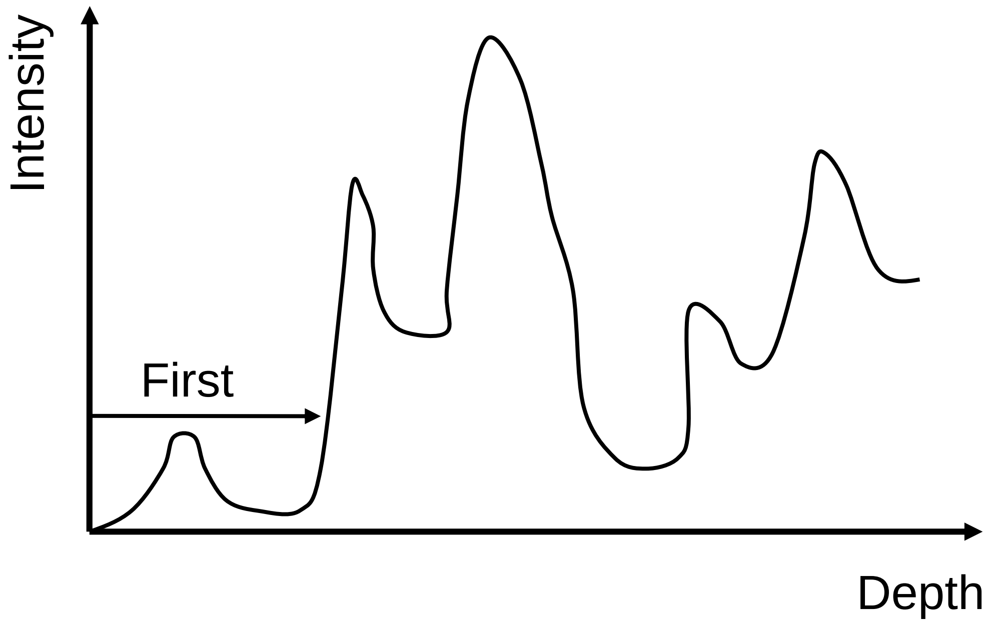


Compositing Schemes



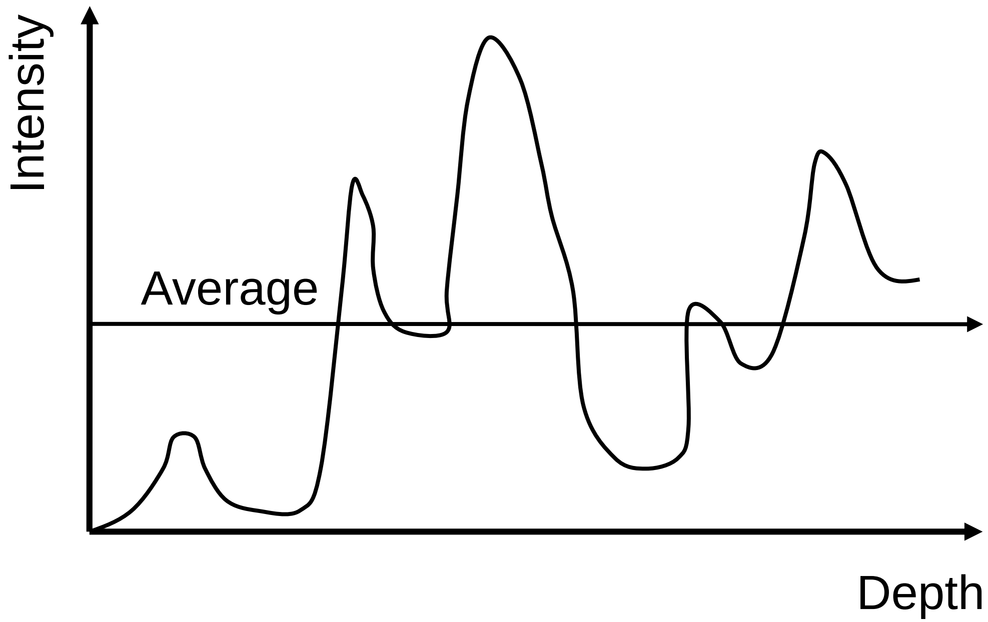
Compositing Schemes

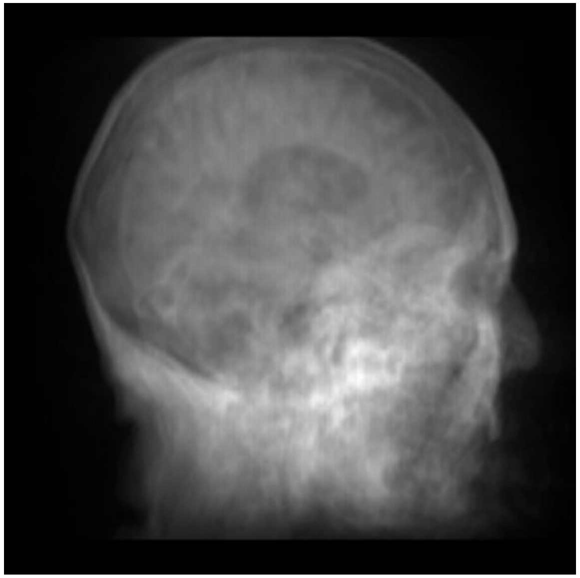
- Compositing: First
- Extracts isosurfaces



Compositing Schemes

- Compositing: Average
- Produces basically an X-ray picture

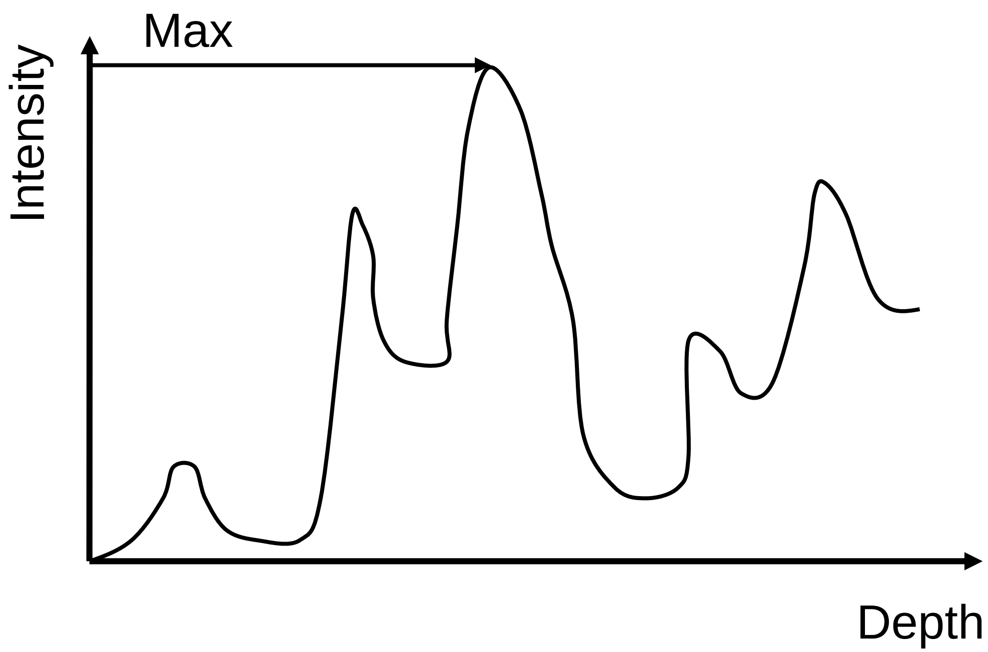


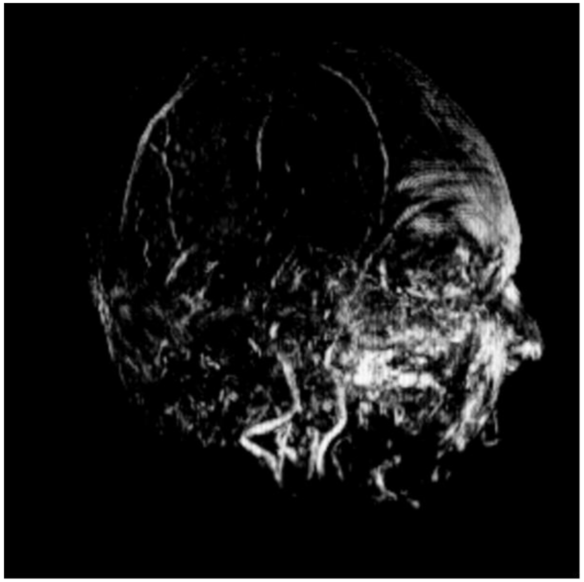


Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Compositing Schemes

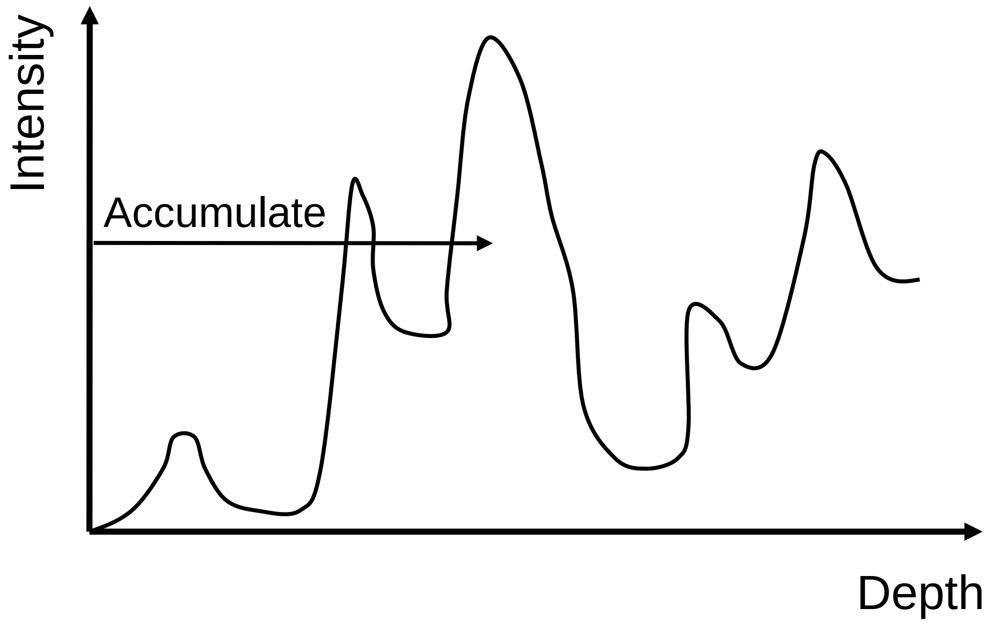
- Maximum Intensity Projection (MIP)
- Often used for magnetic resonance angiograms
- Good to extract vessel structures

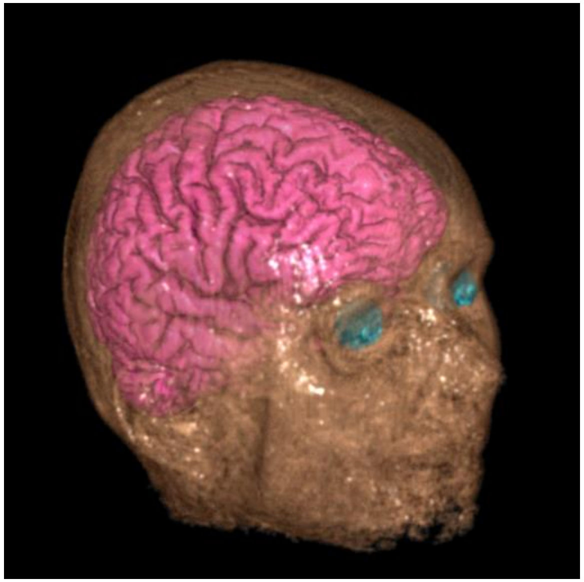




Compositing Schemes

- Compositing: Accumulate
- Emission-absorption model
- Make transparent layers visible (- classification)





Volume Shading

- Lighting / Shading
  - Phong model, Gouraud shading
  - Estimate normal by $N(x,y,z) = \dfrac{\nabla F(x,y,z)}{\left\|\nabla F(x,y,z)\right\|}$
  - Approximate $\nabla F(x,y,z)$ by *central difference*

$$\nabla F(x,y,z) = \frac{1}{2}\begin{pmatrix} F(x_{i+1},y_i,z_i) - F(x_{i-1},y_i,z_i) \\ F(x_i,y_{i+1},z_i) - F(x_i,y_{i-1},z_i) \\ F(x_{i+1},y_i,z_i) - F(x_i,y_i,z_{i-1}) \end{pmatrix}$$

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

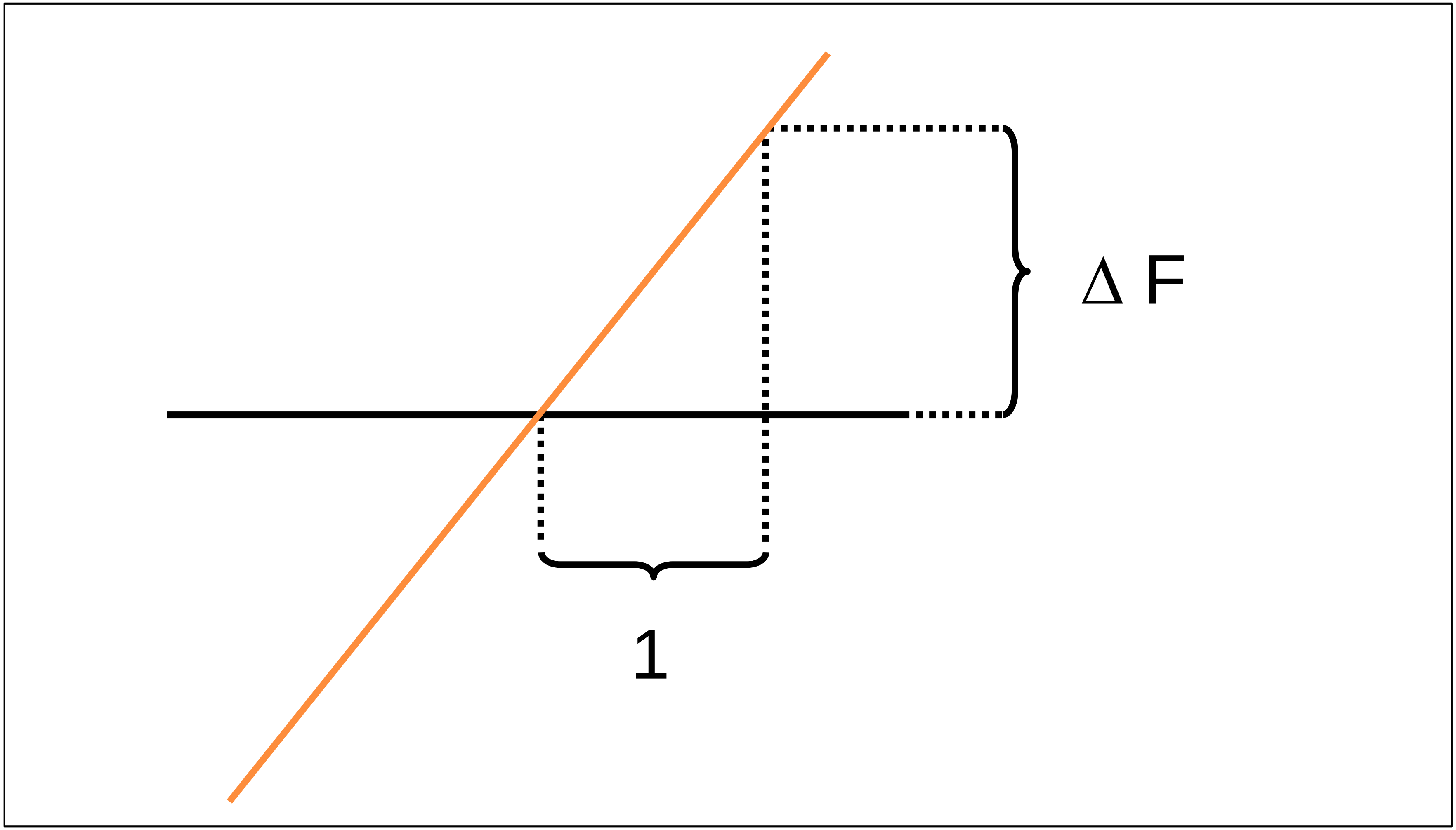
Volume Rendering Modes

- Classification
  - assign spatial opacity values $\alpha(x_i, y_i, z_i)$
  - assign α_v for isovalues $F(x,y,z) = v$ (transfer function)
  - isosurface ... $\alpha_v = \delta(v=v_0)$ ... blur with radius r

$$\alpha(x_i) = \alpha_v \cdot \begin{cases} 1, & \|\nabla F(x_i)\| = 0 \wedge F(x_i) = v \\ 1 - \frac{1}{r} \frac{F(x_i) - v}{\|\nabla F(x_i)\|}, & \|\nabla F(x_i)\| > 0 \wedge \|F(x_i) - v\| < \|\nabla F(x_i)\| \\ 0, & \text{otherwise} \end{cases}$$

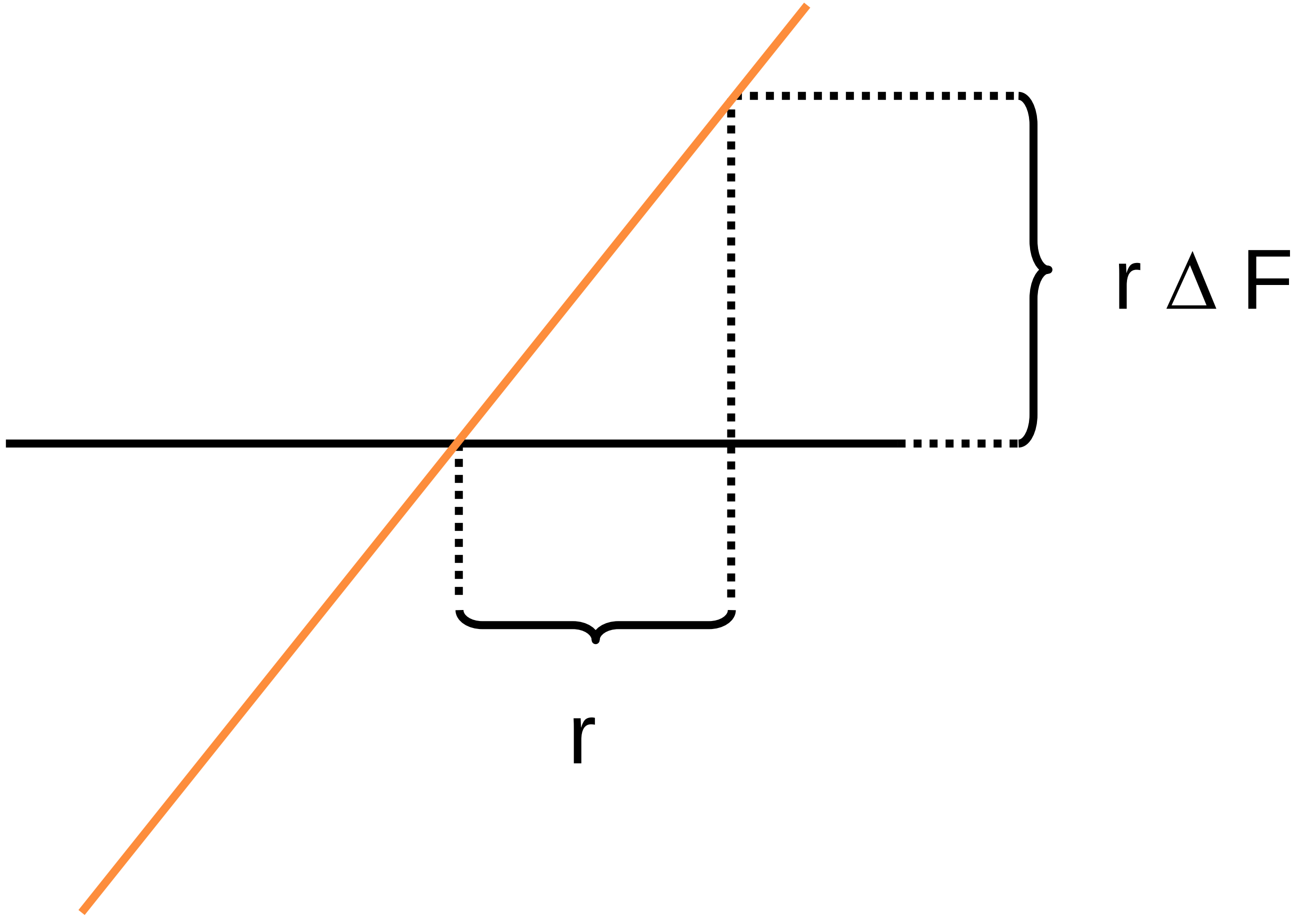
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volume Rendering Modes



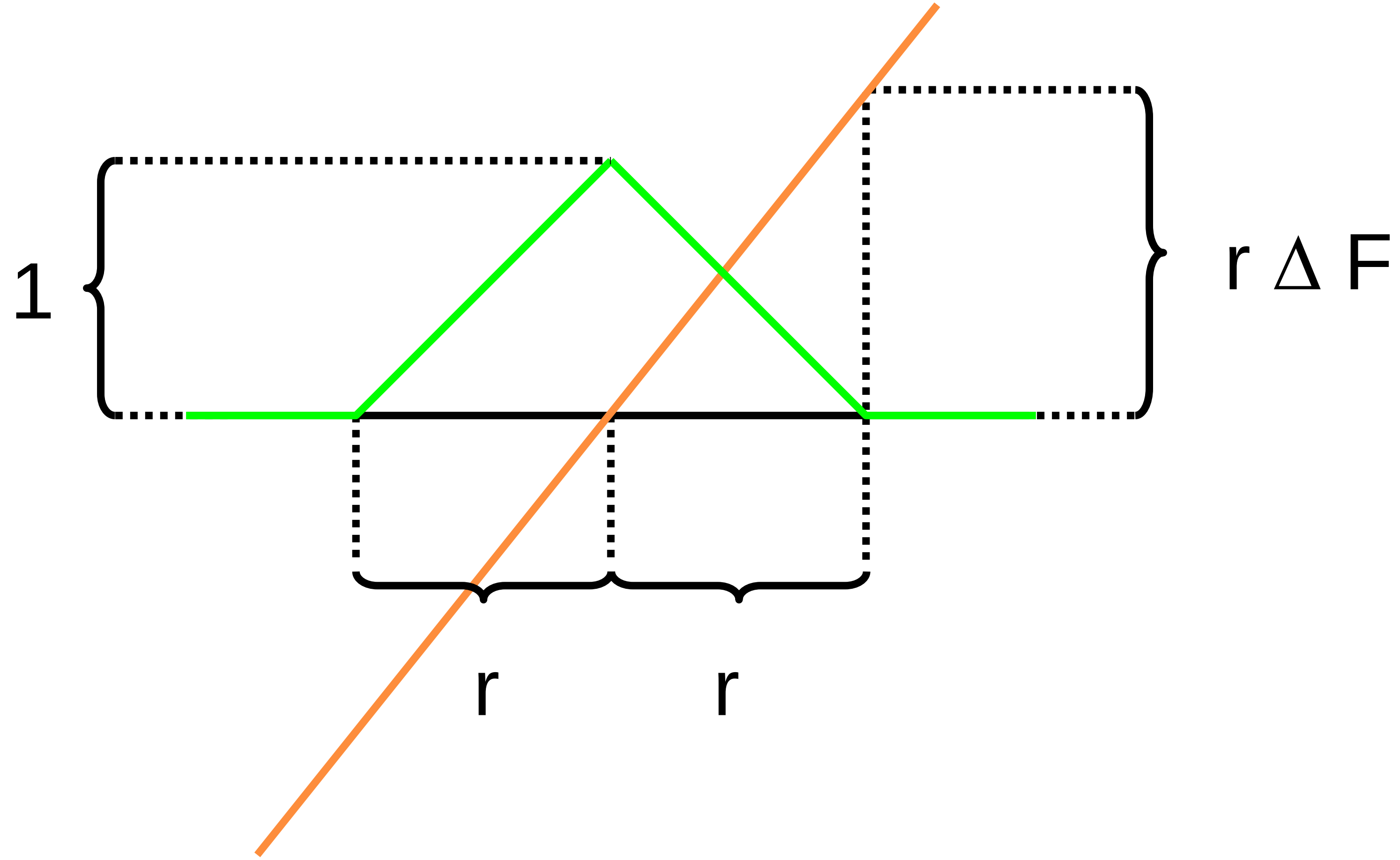
Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volume Rendering Modes



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Volume Rendering Modes



Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization

Ray Casting

- Acceleration techniques
  - Spatial data structures (octree)
    - Find and skip empty regions
    - Find homogenous regions, use lower sample rate
  - Early ray termination
    - Opacity threshold terminates ray traversal
    - Requires front-to-back traversal
  - Fast cell traversal (Bresenham-like)

Visual Computing Institute | Prof. Dr. Leif Kobbelt
Computer Graphics and Multimedia
Data Analysis and Visualization